Broadview®
www.broadview.com.cn

On Software Development

软件开发这点事儿

——软件开发工具手册

邵志东 著

超值附送 55小时教学视频

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

VISIT…

LANZAROTE Caliente.com

软件开发这点事儿——软件开发工具手册

书名：
社名：
开本：
文前页码：
日期：
主管签字：
责编：
校次：
正文页码：
排版员：
排版公司：

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

你还在为软件开发工具的不断更新换代苦恼吗？你还在面对项目无从下手吗？你想成为软件开发的管理者吗？本书会给你所有的答案。本书主要介绍软件开发过程中需要掌握的各种技能和知识，是所有从事软件开发人员的必备工具手册。本书提供给读者三方面内容：从事本项目所涉及的专业技能图、项目实施流程图、项目开发指导图（导做图）；项目实施过程所涉及的知识、技能的归纳、总结和经验分享；大量实际应用模版。生命周期长。同时以 ASP.NET 为例，介绍了一些常用项目的设计、开发过程，非常具有实用性。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（ＣＩＰ）数据

软件开发这点事儿：软件开发工具手册 / 邵志东著.
北京：电子工业出版社，2009.2
ISBN 978-7-121-08072-2

I. 软… II.邵… III.软件工具－手册 IV.TP311.56-62

中国版本图书馆 CIP 数据核字（2008）第 212410 号

责任编辑：李冰
印　　刷：
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路 173 信箱　邮编 100036
经　　销：各地新华书店
开　　本：185×235　1/16　印张：19.5　字数：402 千字
印　　次：2009 年 2 月第 1 次印刷
印　　数：4000 册　　定价：45.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

目　　录

第 1 篇　知识篇

第 1 章　恭喜你，成为软件程序员 ········ 2

作为一名程序员要掌握哪些专业技能？应该具备哪些素质技能？个人的职业发展道路应该如何设计？这是每个程序员都要面对的问题。在本章中，我们将就这几个方面给大家做介绍。

1.1　软件程序员专业技能 ········ 2
1.1.1　专业技能检核表 ········ 2
1.1.2　软件程序员岗位技能图 ········ 4
1.2　软件程序员素质技能 ········ 4
1.2.1　软件程序员的素质技能 ········ 4
1.2.2　高级软件程序员的素质技能 ········ 5
1.3　软件程序员职业发展图 ········ 6
1.3.1　软件企业的人才类型和素质结构关系 ········ 6
1.3.2　高级技术人才理想成长路径 ········ 7
1.3.3　高级管理人才理想成长路径 ········ 7
1.3.4　高级综合人才理想成长路径 ········ 8
第 2 章　软件开发流程 ········ 9

什么是产品？什么是项目？它们的开发流程有何不同？如何基于项目进行产品开发？如何基于产品做项目开发？软件开发中，质量和效率如何平衡？这些都是每个软件开发人员面临的问题。在本章中，我们将共同来探讨这些话题。

2.1　项目还是产品——程序员心中的痛 ········ 9

2.2 软件开发流程 …… 10
2.2.1 项目开发流程 …… 10
2.2.2 产品开发流程 …… 11
2.3 标准还是效率——痛苦的抉择 …… 11
2.3.1 质量/过程标准的来源 …… 11
2.3.2 标准能做到什么和不能做到什么 …… 12
2.3.3 质量标准和过程控制有什么用 …… 12
2.3.4 软件开发过程中如何应用这些标准 …… 13
第 3 章 可行性研究 …… 14

可行性分析是要决定“做还是不做”。显然，“做还是不做”比“如何做”更重要。如果可行性分析不够充分、完整，就有可能得出错误的结论，从而导致资金的浪费、无谓的投入。因此可行性分析至关重要。在本章中我们将学习如何从市场、政策、技术、经济，以及 SWOT 方面做可行性分析。

3.1 可行性分析的要素 …… 14
3.2 市场可行性分析 …… 15
3.3 政策可行性分析 …… 15
3.4 技术可行性分析 …… 15
3.5 成本—收益分析 …… 16
3.6 SWOT 分析 …… 16
3.6.1 竞争优势分析 …… 17
3.6.2 竞争劣势分析 …… 17
3.6.3 外部机会分析 …… 18
3.6.4 外部威胁分析 …… 18
3.6.5 SWOT 核心问题 …… 19
第 4 章 需求分析 …… 20

需求分析是要决定“做什么，不做什么”。在用户导向的今天，用户需求分析变得越来越重要。在需求分析阶段，我们需要完成哪些工作？常用的需求分析工具有哪些？在本章中，这些问题会一一得到解答。

4.1 需求分析的重要性 …… 20
4.2 需求分析为什么困难 …… 20

4.3 需求分析基本任务 21
4.4 需求分析的结果 22
4.5 需求分析阶段工作重点 22
4.6 UML 和 Rational Rose 22
4.7 Microsoft Visio 23
第 5 章 系统设计 24

每个程序员都希望自己能够成为系统架构师。要想成为一名系统架构师，充分了解系统设计的相关知识、掌握系统设计的相关工具就显得非常重要。本章中，我们将就如何进行系统设计进行详细介绍。

5.1 系统设计工作流程图 24
5.2 面向对象的设计方法 25
5.2.1 面向对象的概念 26
5.2.2 面向对象设计方法目标 29
5.3 三层应用程序模型 29
5.4 函数、对象和组件 30
5.4.1 函数和过程 30
5.4.2 类和对象 30
5.4.3 组件 30
5.5 系统设计内容 31
5.5.1 数据库设计 32
5.5.2 界面设计 42
5.5.3 模块设计 46
5.5.4 数据结构和算法设计 48
5.5.5 OOP（Object Oriented Programming）设计方法步骤 49
第 6 章 编码 50

作为一名程序员，首先要精通一种面向对象的编程语言，熟练掌握相应的开发工具；随着网络技术的飞速发展，有很多的网络编程技术如 XML、Ajax 等都需要我们去了解和学习；作为项目中的一员，了解编码规范和源代码控制工具也是程序员所必须掌握的。在本章中，我们将探讨这些话题。

6.1 编程语言和开发工具 50

6.1.1 编程语言 ········ 50
6.1.2 开发工具 ········ 52
6.2 网络编程必备知识 ········ 53
6.2.1 HTML/JavaScript ········ 53
6.2.2 XML ········ 53
6.2.3 WebService ········ 54
6.2.4 Ajax ········ 54
6.3 了解极限编程 ········ 55
6.4 编码规范 ········ 56
6.5 源代码控制 ········ 56
6.5.1 版本管理的必要性 ········ 56
6.5.2 常用源代码控制工具 ········ 57
6.5.3 源代码控制规范 ········ 58

第 7 章 软件测试 ········ 59

在某些开发人员眼里，测试很神秘；在某些开发人员眼里，测试很简单；在某些开发人员眼里，测试就是质量。那么软件测试到底是什么？作为程序员来说，和软件测试人员相比，应该了解和掌握哪些方面的测试知识？在本章中我们将着重介绍这些方面的知识。

7.1 软件测试基本概念 ········ 59
7.1.1 单元测试 ········ 60
7.1.2 集成测试 ········ 60
7.1.3 系统测试 ········ 62
7.1.4 性能、压力和负载测试 ········ 62
7.1.5 测试成本 ········ 63
7.1.6 黑盒测试和白盒测试 ········ 63
7.2 白盒测试用例 ········ 64
7.2.1 测试用例简介 ········ 64
7.2.2 白盒测试用例设计 ········ 66
7.2.3 测试用例设计步骤 ········ 67
7.3 软件测试工具 ········ 73
7.3.1 常见软件测试工具列表 ········ 73

7.3.2　常用单元测试工具介绍······74

第 8 章　结项和产品化······76

编码结束了，软件开发就结束了吗？不！远远没有，还有很多工作要做。制作安装程序、进行软件加密、编写说明书和系统帮助文档，都是在结项和产品化阶段要做的工作。

8.1　安装盘制作······76
8.1.1　使用 InstallShield······76
8.1.2　使用 Visualstudio.NET 发布程序······76
8.1.3　安装盘制作规范······77
8.2　软件加密······77
8.2.1　软件加密概述······77
8.2.2　常见的软件加密技术······77
8.3　说明书制作······79
8.3.1　Word 书籍版说明书······79
8.3.2　电子说明书制作······79

第 9 章　项目组建设······81

随着软件编程技术的发展，团队开发变得越来越普遍，因此如何能够提高整个团队的协作能力、提高整体的开发效率，是每个软件企业都关注的问题。在本章中，我们将探讨有关项目建设方面的问题：项目守则、公共代码积累、工作计划总结等。

9.1　项目组守则······81
9.2　总结常用代码······81
9.3　共享编程技巧······82
9.4　编写工作计划总结······82
9.5　代码量统计工具······83

第 2 篇　工具篇

第 10 章　Rational Rose 实例入门······86
10.1　创建工程······86
10.2　建立 Actor······88
10.3　创建 Use Case······89

10.4 用顺序图描述 Use Case 90
10.5 建立协作图 92
10.6 建立类 93
10.7 实现模型 94
10.8 UML 符号和图形 95

第 11 章 Power Designer 实例入门 99

11.1 创建项目工程 99
11.2 建立概念数据模型 100
11.2.1 创建实体 100
11.2.2 添加属性 101
11.2.3 定义属性的标准检查约束 102
11.2.4 定义实体的主、次标识符 103
11.2.5 创建其他实体 105
11.2.6 建立联系 107
11.3 生成物理数据模型 110
11.4 生成数据库建库脚本 112
11.5 生成关系数据库 113
11.6 Power Designer 的 4 种模型文件 113

第 12 章 NUnit 实例入门 115

12.1 NUnit 下载安装 115
12.2 创建测试项目 116
12.3 添加 NUnit 引用 116
12.4 创建测试用例 117
12.5 创建被测试类 118
12.6 编译项目 118
12.7 运行图形测试工具 118
12.8 修改测试 119
12.9 异常测试 120

第 13 章 JUnit 实例入门 121

13.1 JUnit 下载和安装 121
13.2 创建测试项目 122

13.3 添加 JUnit 引用……122
13.4 创建被测试类……124
13.5 创建测试用例……125
13.6 创建测试单元……126
13.7 运行测试……127
13.8 JUnit 结构说明……128

第 14 章 InstallShield 实例入门……129

14.1 创建安装项目……129
14.2 安装程序语言设计……130
14.3 应用程序信息设置……131
14.4 设置程序运行条件……131
14.5 设定安装架构……131
14.6 添加要发布的网站文件……131
14.7 添加程序快捷键……132
14.8 添加注册表修改……133
14.9 安装界面设置……134
14.10 多语言字符串修改……134
14.11 生成安装……134
14.12 运行安装文件……135
14.13 添加数据库支持……136
14.14 直接附加数据库……141
14.15 程序卸载……141

第 15 章 “VisualStudio.NET 部署应用程序”实例入门……142

15.1 创建要发布 ASP.NET Web 应用程序……142
15.2 创建安装程序类……143
15.3 创建包含 SQL 语句的文本文件以创建数据库……143
15.4 将代码添加到安装程序类中以阅读文本文件……145
15.5 创建部署项目……147
15.6 将主程序项目的输出添加到部署项目中……148
15.7 创建自定义安装对话框……148
15.8 创建自定义操作……149

15.9 程序部署 ········ 150

第 16 章 HLP 文件制作实例入门 ········ 151

16.1 RTF 源文件制作 ········ 151
16.2 编译成为 HLP 文件 ········ 155
16.3 制作主题帮助文件（CNT 文件） ········ 158
16.4 把 CNT 文件加入工程 ········ 159

第 17 章 CHM 文件制作实例入门 ········ 161

17.1 材料准备 ········ 161
17.2 创建工程 ········ 161
17.3 设置 CHM 文档的标题 ········ 164
17.4 修改文档对话框属性 ········ 167
17.5 制作目录 ········ 168
17.6 添加目录 ········ 170
17.7 制作索引 ········ 172
17.8 添加索引 ········ 173
17.9 说明 ········ 174

第 18 章 Visual SourceSafe 实例入门 ········ 175

18.1 软件安装 ········ 175
18.2 服务端创建 VSS 数据库和用户 ········ 175
18.3 客户端选择数据库 ········ 177
18.4 添加项目到 VSS 数据库 ········ 178
18.4.1 通过 SourceSafe 添加项目 ········ 179
18.4.2 通过 VisualStudio.Net2005 集成开发环境添加项目 ········ 180
18.5 在客户端 VS.NET 开发环境中使用 VSS ········ 182
18.6 修改历史查看 ········ 184
18.7 用户权限修改 ········ 185

第 19 章 CVS 实例入门 ········ 187

19.1 安装 CVSNT ········ 187
19.2 服务端创建仓库和用户 ········ 189

19.3 添加项目到 CVS 服务器 ······ 189
19.4 在 Eclipse 客户端开发环境中使用 CVS ······ 193
19.5 VS.NET 中集成 CVS ······ 196
19.6 CVS 和 VSS 比较 ······ 199

第 20 章 Visio 实例入门 ······ 200

20.1 建立新文档 ······ 200
20.2 创建流程图 ······ 201
20.3 保存 ······ 204
20.4 自定义形状的制作 ······ 204

第 3 篇 实例篇

第 21 章 网上商城系统开发实战 ······ 208

21.1 项目概述 ······ 208
21.2 可行性分析 ······ 209
21.2.1 系统目标 ······ 209
21.2.2 开发的可行性分析 ······ 209
21.3 需求分析 ······ 210
21.3.1 确定用户类型 ······ 210
21.3.2 用户工作流程 ······ 210
21.3.3 软件功能 ······ 212
21.3.4 UML 图 ······ 213
21.3.5 系统运行环境 ······ 214
21.4 系统设计 ······ 215
21.4.1 数据库设计 ······ 216
21.4.2 页面设计 ······ 228
21.4.3 页面关系图 ······ 230
21.4.4 界面设计 ······ 231
21.5 编码和测试 ······ 244
21.5.1 公共模块设计 ······ 244
21.5.2 源代码控制 ······ 254
21.5.3 单元测试工具 ······ 254
21.5.4 系统源代码 ······ 254

21.6 制作安装盘……254
21.7 系统源文件……254

第 4 篇 模板与规范篇

第 22 章 软件开发模板……256

在本章中，我们提供了一套简化后的质量模板文件，比较适合中小型软件企业的应用。

22.1 软件立项申请书……257
22.2 可行性分析报告……259
22.3 立项评审报告……260
22.4 用户需求说明书……261
22.5 需求分析评审报告……265
22.6 总体设计书……266
22.7 开发策划书……266
22.8 详细设计书……268
22.9 编码计划……268
20.10 测试方案……269
20.11 测试报告……270
22.12 软件开发结项报告……272

第 23 章 项目组规范……273

在本章中，我们提供了团队开发所需要的各种开发规范，包括项目组工作守则、编码规范、源代码控制规范、界面设计规范、数据库使用规范、安装盘制作规范等，会极大地方便团队开发。

23.1 项目组工作守则……273
23.2 C#编码规范……276
23.2.1 命名……277
23.2.2 代码格式书写规范……278
23.2.3 其他……280
23.3 SourceSafe 使用规范……281
23.4 数据库命名规范……283

23.5　用户界面设计规范 286
23.6　安装盘制作规范 290
后记 291

第1篇 知识篇

本篇导读

软件技术的发展日新月异,在快速的技术变迁中,程序员如何能够一直保持在技术的前沿?程序员岗位真是一个高淘汰率的职位吗?如何在快速的变化中把握软件编程的核心?笔者认为:了解软件开发的过程,把握各个阶段需要掌握的基本知识,熟悉各个阶段所涉及的工具,这才是“以不变应万变之道”!所以在本篇中我们将重点讲解软件开发各个阶段作为一个程序员需要掌握的知识。

章　　节	标　　题	基本内容
第1章	恭喜你,成为软件程序员	• 软件程序员应该具备的专业技能、素质技能 • 了解软件程序员一般职业发展规律
第2章	软件开发流程	• 认识软件产品和软件项目 • 了解质量标准和过程控制的作用
第3章	可行性研究	• 介绍软件开发各个阶段需要掌握的知识点、技能和工具
第4章	需求分析	
第5章	系统设计	
第6章	编码	
第7章	软件测试	
第8章	结项和产品化	
第9章	项目组建设	• 了解软件开发组需要遵守的一些规范 • 了解开发组日常工作中需要做的积累工作。

第 1 章　恭喜你，成为软件程序员

本章导读：作为一名程序员要掌握哪些专业技能？应该具备哪些素质技能？个人的职业发展道路应该如何设计？这是每个程序员都要面对的问题。在本章中，我们将就这几个方面给大家做个介绍。

1.1　软件程序员专业技能

1.1.1　专业技能检核表

首先要恭喜你已经成为或有志成为一名软件程序员！

软件程序员不同于编码员，软件开发作为一个过程，除了编码以外，还包括很多的环节；作为程序员也需要了解其他很多的知识。

下面是一个检核表，大家可以对照一下，看自己在软件开发方面掌握技能的情况。

软件程序员专业技能自检表

基础项		是否熟练掌握	是否有项目经验	是否列入学习计划
操作系统	Windows	□	□	□
	Linux	□	□	□
	UNIX	□	□	□
	其他请注明	□	□	□

续表

基础项		是否熟练掌握	是否有项目经验	是否列入学习计划
数据库	SQL Server	☐	☐	☐
	ORACLE	☐	☐	☐
	DB2	☐	☐	☐
	其他请注明	☐	☐	☐
基础知识	HTML 语言	☐	☐	☐
	XML 语言	☐	☐	☐
	数据结构常用算法	☐	☐	☐
	UML 知识	☐	☐	☐
开发语言和工具	VC++	☐	☐	☐
	C++	☐	☐	☐
	VS.NET（VB.NET、C#等）	☐	☐	☐
	Java	☐	☐	☐
	其他请注明	☐	☐	☐
软件过程	软件工程理论	☐	☐	☐
	质量体系和过程控制	☐	☐	☐
	软件设计	☐	☐	☐
	单元测试工具	☐	☐	☐
	软件加密	☐	☐	☐
	源代码控制	☐	☐	☐
	说明书编写	☐	☐	☐
	程序部署	☐	☐	☐
项目组建设	编码规范	☐	☐	☐
	源代码控制规范	☐	☐	☐
	数据库设计规范	☐	☐	☐
	公共组件或基类	☐	☐	☐

测试结果怎么样？如果所列的技能大部分你都已经熟练掌握，那么恭喜你，你是一名合格的程序员了！如果这些技能大部分你都已经拥有了项目经验，那么更要恭喜你，你是一名优秀的程序员了！如果都没有，那么赶快把他们列入你的学习计划，跟随本书的学习，你很快就会成为一名真正的程序员！

1.1.2 软件程序员岗位技能图

对照上面的表格，我们把软件程序员应该掌握的项目知识用框架图表示，如图 1-1 所示。



图 1-1 软件程序员岗位技能图

1.2 软件程序员素质技能

1.2.1 软件程序员的素质技能

除了要具备上述所列的专业技能外，作为一个程序员又应该具备哪些素质方面的技能呢？结合作者本人的经验和一些同仁的共识，我们把这些技能列举如下。

软件程序员素质技能表

	技能项	说　　明
1	写文档和注释的习惯	优秀的程序员是非常注重注释和文档的编写的； 作为程序员，30%的工作时间写注释和技术文档是很正常的，而作为高级程序员和系统分析员，这个比例还要高很多
2	模块复用性思维习惯	软件最大的优点是能够复用，因此程序员在完成任何一个模块或函数时，不要局限在完成当前的任务上，要多想一些，想想看该模块是否可以脱离这个系统存在，是否能够重复使用
3	测试习惯	程序员在编码阶段的测试工作将会大大提高系统的可靠性。任何编码阶段的疏忽，会带来后续阶段几倍、几十倍的解决代价
4	学习、总结的习惯	现代社会，技术的发展日新月异，程序员是很容易落伍的职业。养成学习总结的习惯，程序员才能跟上时代的步伐，不断提升自己
5	团队精神	现代的软件开发往往是由一个团队共同完成的。独立承担自己的工作任务、遵守共同的编码规范、书写规范化的代码、协助其他成员完成工作……都是团队精神的体现
6	沟通能力	在软件开发过程中，不仅程序员之间要进行交流合作，程序员和销售人员、测试人员等各个部门的人、甚至客户都要沟通交流

1.2.2　高级软件程序员的素质技能

每个程序员（初级程序员）都希望自己能够成为一名高级程序员、系统分析员或者系统架构师。要成为一名高级程序员，应该具备哪些素质方面的技能呢？

高级程序员素质技能表

	技能项	说　　明
1	需求分析能力	需求分析是软件开发流程中非常重要的一个环节。对于研发项目的组织和管理者来说，他们不但要理解客户需求，还要具备把用户需求转化为系统设计的能力
2	系统设计能力	高级程序员者必须能够掌握不少于两种到三种的项目设计方法(比如自顶至下的设计方法，比如快速原型法等），并能够根据项目需求和资源搭配来选择合适的设计方法进行项目的整体设计
3	模块分解能力	高级程序员要有能够把整体系统进行模块化分解的能力，同时还要对分解产生模块的复用性进行考虑
4	整体项目评估能力	作为高级程序员，必须能够从全局出发，对项目有整体的清醒认识。包括评估项目整体和各个模块的工作量、评估项目所需的资源等

续表

	技能项	说　　明
5	项目组织能力	包括工作的量化和分配能力、团队协调能力。没有量化就很难做到合适的绩效考核，而程序量化又不是简单的代码行数可以计算的，因此要求高级程序员能真正评估一个模块的复杂性和工作量。项目组内程序员的水平是有差距的，同时用户需求又在不断变化，高级程序员要有动态调整责权和成员工作状况的能力

1.3 软件程序员职业发展图

1.3.1 软件企业的人才类型和素质结构关系

在软件企业当中，作为一名程序员，也可以从两个方向去发展：

- 技术方向：作为一个刚入门的程序员，你可以沿着“编程技术→测试技术→软件设计技术→需求分析技术→系统架构技术”去发展。
- 管理方向：首先是个人的基本素质修养，然后可以沿着“时间管理→风险管理→计划管理→过程管理→成本管理→人际沟通→领导能力”的方向去学习锻炼一些管理素质方面的技能。其实随着职位的提升，对以上管理技能掌握的要求会越来越高。

上述内容可以用图 1-2 来表示。

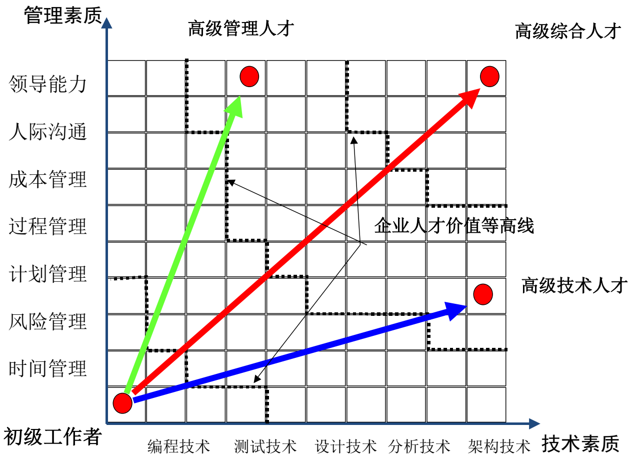


图 1-2　软件企业人才发展路径

1.3.2　高级技术人才理想成长路径

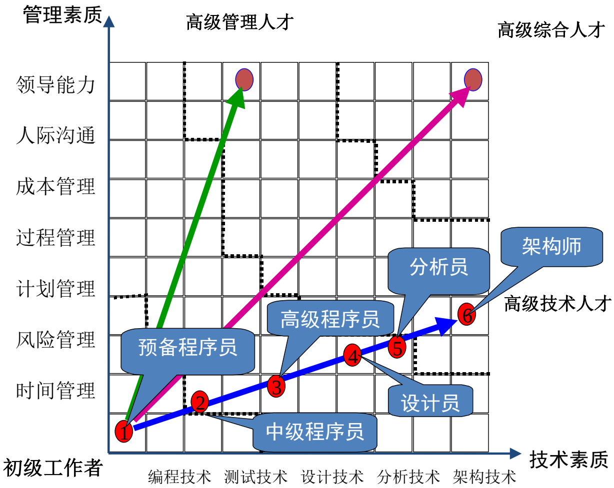


图 1-3　高级技术人才理想成长路径

注意：虽然以上是技术人才的成长路径，但也可以看到随着职业的提升，对各种管理技能的要求也越来越高。因此高级技术人才并不仅仅是“技术高手”，也是“管理能手”。

1.3.3　高级管理人才理想成长路径

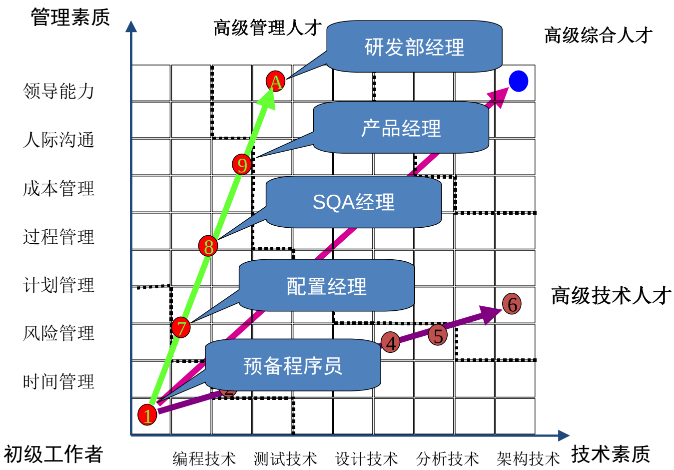


图 1-4　高级管理人才理想成长路径

由于软件企业的特殊性，往往很多管理人员都是从技术出身的。依靠本身的技术背

景，努力提升自己的管理素质和能力，再加上一些机遇，你就可以成为一名管理型人才。

1.3.4　高级综合人才理想成长路径

要说当今社会，什么样的人才最抢手？那首先要说是一些复合型人才！在软件企业中，既有深厚的技术底蕴，又有较强的管理能力的综合性人才同样是大受欢迎的。

图 1-5 画出了高级综合人才的成长路径，为有志成为一名综合性人才的朋友们指出了个人职业发展方向！

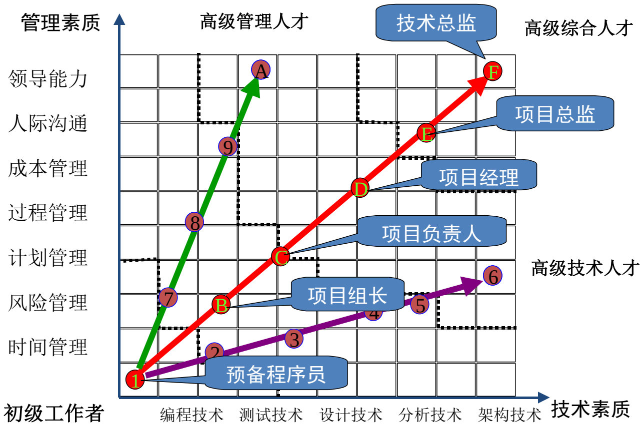


图 1-5　高级综合人才理想成长路径

第 2 章　软件开发流程

本章导读：什么是产品？什么是项目？它们的开发流程有何不同？如何基于项目进行产品开发？如何基于产品做项目开发？软件开发中，质量和效率如何平衡？这些都是每个软件开发人员面临的问题。在本章中，我们将共同来探讨这些话题。

2.1 项目还是产品——程序员心中的痛

软件最大的特点就是容易复制，并且复制的成本接近为零。在软件行业，所有的软件厂商都希望自己开发出的软件能够被所有的目标客户使用，从而达到“一次开发，重复使用”的目的。典型的代表是微软公司，其所有开发的软件都是面向众多的普通客户，一般称为软件产品。产品的用户一般是大量的规模用户，一般不会为某个用户而个性化定制修改软件，而主要满足大多数用户的需求。比如，“我希望 Windows 的开始菜单默认能够在屏幕中间”这样的特殊要求，微软公司一般是不予理会的。

但在现实情况中，往往有些组织希望软件公司为其提供定制开发软件，这种软件一般称为项目软件。它主要是为满足某个或某些特定用户的。

当然，如果提供足够的回报，软件产品开发商也会把其产品为某个用户进行个性化修改，我们称之为“基于产品的项目开发”。

另外，一些行业软件开发商，往往是从项目开发开始的，但也非常希望能够发展成为整个行业的通用软件产品。比如“用友财务软件”的发展历程就是这样的，我们称之为“基于项目的产品开发”。

下面我们用一个表格来对“项目”和“产品”做一个对比。

项目和产品

	用　户	特　点		典型代表
项目	某个企事业单位	需求来自于某个用户	定制；多为一次性卖出	某企业的内部办公系统
产品	某一类企事业单位	需求来自于某类用户	无法为某个用户定制；可以重复复制销售，充分体现软件的特点	Windows，用友财务软件
基于项目的产品开发	是指以某些同类的项目为基础，最终发展成为产品。这是所有软件厂商的目标			
基于产品的项目开发	是指以成熟产品为基础，针对某个有实力的用户进行个性化定制			

2.2 软件开发流程

2.2.1 项目开发流程

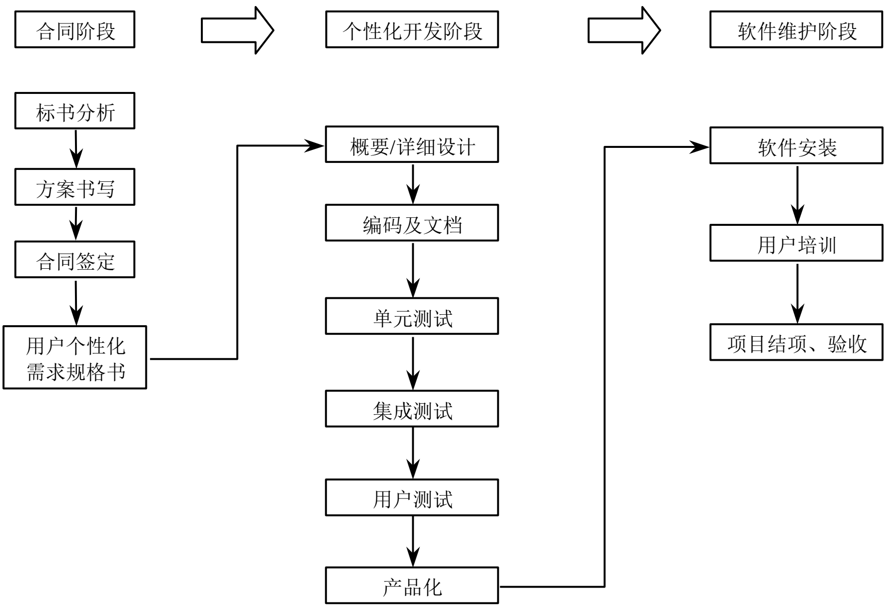


图 2-1　项目开发工作流程

2.2.2 产品开发流程

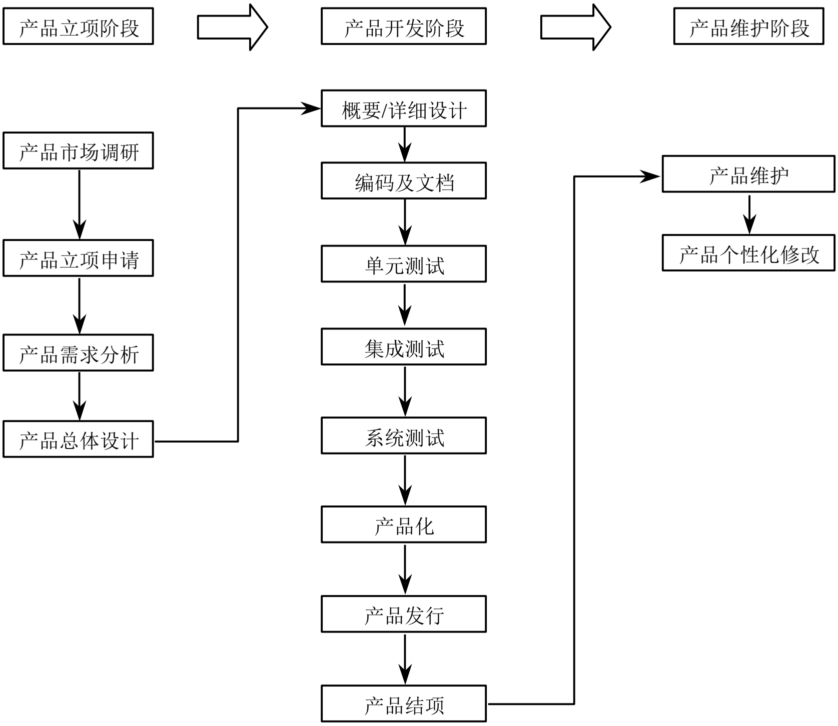


图 2-2 产品开发流程

2.3 标准还是效率——痛苦的抉择

对于软件人员来说，往往严格执行质量体系的代价就是效率的降低，到底是执行标准还是要提高效率？的确是一个痛苦的选择，但同时这也是一个平衡的问题，需要根据现状进行取舍。

2.3.1 质量/过程标准的来源

开始并没有什么质量标准或者过程标准，但有些组织和企业呢，做得很成功，而有些则不成功。于是就有人就去分析为什么这些组织和企业成功了呢？他们有哪些共同的特征吗？答案是有。这些特征被归纳出来，并应用管理理论的成果，使之成为一种体系，就诞生了 ISO9000、CMM 等标准和体系。

2.3.2　标准能做到什么和不能做到什么

标准是客观存在的东西，它是否能起到作用，关键还是要看操作的人。

如果操作者有意识，标准就可以帮忙。比如你没有做可行性分析，SQA 就会提醒，这违背了规程，迫使你按照标准操作。

但是如果 SQA 没有意识提醒，或者即使提醒了你仍然没有进行修改，那即使有标准，标准也是无能为力的。

2.3.3　质量标准和过程控制有什么用

为什么要提倡质量标准和进行过程控制呢？为什么对于软件企业这些工作更为重要？到底这些工作会有什么用呢？

经过反复的研究和总结，笔者认为，对于软件企业，质量标准和过程控制最突出的有以下一些效用：

质量标准和过程控制的作用

	作　　用	说　　明
1	可控制	按照质量标准，在软件开发过程的每个关键阶段，都需要进行讨论和评审，这样使得整个过程是可控制的，降低了软件开发的风险
2	可追溯	所有的质量文件都要求按照标准保存，这样即使项目过去了很长时间，也可随时追溯查询原软件开发过程中各个阶段的重要事件
3	降低人员流动的危害	软件企业是知识型的企业，最大的成本就是人力成本，最大的危害就是人员流动带来的知识的损害。有了质量控制，可以让知识和开发经验以文字的形式得以积累，降低了人员流动所带来的危害
4	专业化形象	在现代社会，企业推行质量体系已经成为塑造企业专业化形象最重要的一环。国内很多企业推行 ISO、CMM 就是一个典型验证

从以上表格可以看出，质量体系和标准非常重要。作为一名程序员，必须了解质量体系相关内容，并在软件开发过程中贯彻质量体系的一些具体措施。

2.3.4 软件开发过程中如何应用这些标准

根据企业的规模、项目组的大小，在效率和标准之间进行平衡。一般来说，企业规模越大，用在标准和过程控制上所花的精力就应该越大。但即使是小企业，也应该保留一些基本的过程控制项。在本书模板篇中，我们提供了一套简化的开发模板，它包括“用户需求规格说明书”、“需求评审报告”、“系统设计书”、“系统开发进度计划”、“项目验收标准”、“项目结项报告”等文档，特别适合中小型软件企业应用。在本书的配套光盘中，我们提供了这些文档的电子版本。

第 3 章　可行性研究

本章导读：可行性分析是要决定“做还是不做”。显然，“做还是不做”比“如何做”更重要。如果可行性分析不够充分、完整，就有可能得出错误的结论，从而导致资金的浪费、无谓的投入。因此可行性分析至关重要。在本章中我们将学习如何从市场、政策、技术、经济，以及 SWOT 方面做可行性分析。

3.1 可行性分析的要素

做可行性分析不能以偏概全，也不可以什么鸡毛蒜皮的细节都加以权衡。可行性分析必须为决策提供有价值的证据。

一般地，可行性分析的要素有：

- 市场可行性分析
- 政策可行性分析
- 技术可行性分析
- 成本－收益分析
- SWOT（强项、弱项、机会、威胁）分析

3.2 市场可行性分析

进行市场分析，首先要充分了解本软件所处的整个市场环境。如果整个市场环境已经处于消亡期，盲目地进入市场，肯定会损失惨重。如果市场已经成熟，则相对竞争要激烈一些，那么对本软件的要求就要高，这时候就要详细分析一下本软件在成本方面、功能特性方面是否具有明显优势；如果是潜在市场，则要准确分析整个市场的预计容量，本软件可能的市场占有率，这种情况下，风险相对要高，同时潜在的利润也大。

其次要进行同类竞争对手分析，包括对手软件的市场占有、定价情况、系统功能特点等，做到知己知彼。

最后，要了解最终用户，即消费者的消费方式、用户喜好等，毕竟他们是真正的买家。

3.3 政策可行性分析

要了解对于相关市场，是否有政策扶持或者限制？这点尤为重要。有政策扶持，会有事半功倍的效果；有政策限制，则要对相关政策做深入分析，并提前做必要准备。

3.4 技术可行性分析

技术可行性分析可以简单地表述为：做得了吗？做得好吗？做得快吗？

技术可行性分析至少要考虑以下几方面因素：

（1）是否有核心关键技术无法实现？这点至关重要。用户的要求可能是异想天开的，但如果没有相应的技术能实现，则技术上就是不可行的。

（2）在给定的时间内是否能够完成所有功能？整个软件开发是有时间限制的，若未能在规定的时间内完成，轻则拖延进度，重则断送整个项目。

（3）是否对软件的性能和效率有特殊要求，这些要求能否实现？不同的客户对软件系统的性能和效率要求不同，有些应用如一些网络应用，特别强调系统的并发能力；有些应用强调给客户的反应时间；有些应用如银行客户对安全性要求特别高。这些在开发之前都要做深入考虑。

（4）软件的生产率如何？如果生产率低下，能赚到的钱就少，并且会逐渐丧失竞争力。在统计软件总的开发时间时，不能漏掉用于维护的时间。软件维护是非常拖后腿的事，它能把前期拿到的利润慢慢地消耗光。

3.5 成本—收益分析

企业的首要目标是利润，因此进行成本—收益分析也显得非常重要。

如果软件是项目，则整体收入要在客户合同中确定；如果是产品，那么收入就是销售额。

企业在产品开发、营销、维护方面所付出的都可以折算成金钱，汇总起来就是总成本。要考虑的成本有：

- 软件开发人员与管理人员的工资和福利。
- 办公室房租。
- 办公费用（包括办公耗材、水电费、打印复印费等）。
- 固定资产折旧费用（如计算机、打印机、网络等设备）。
- 通信费。
- 办公消耗，如水电费、打印复印费等。
- 软件开发人员与行政人员的工资。
- 购买系统软件的费用，如买操作系统、数据库、软件开发工具等。
- 市场费（包括广告费、Web 站点建设费、交际费用等）。
- 公司人员培训费用。
- 产品售后服务、维护费

3.6 SWOT 分析

SWOT 分别代表：Strengths（优势）、Weaknesses（劣势）、Opportunities（机会）、Threats（威胁）。SWOT 分析通过对优势、劣势、机会和威胁加以综合评估与分析得出

结论，然后再调整企业资源及企业策略，来达成企业的目标。

SWOT 分析已逐渐被许多企业运用到包括：企业管理、人力资源、产品研发等各个方面，也是软件立项过程中最常用、非常有效的可行性分析方法。

3.6.1 竞争优势分析

竞争优势（S）是指一个企业超越其竞争对手的能力，或者指公司所特有的能提高公司竞争力的东西。

竞争优势可以是以下几个方面：

- 技术技能优势：独特的技术，低成本运营方法，领先的革新能力，雄厚的技术实力，完善的质量控制体系，丰富的营销经验，优秀的客户服务，卓越的大规模采购技能。
- 有形资产优势：先进的生产流水线，现代化车间和设备，拥有丰富的自然资源储存，吸引人的不动产地点，充足的资金，完备的资料信息。
- 无形资产优势：优秀的品牌形象，良好的商业信用，积极进取的公司文化。
- 人力资源优势：关键领域拥有专长的职员，积极上进的职员，较强的组织学习能力，丰富的经验。
- 组织体系优势：高质量的控制体系，完善的信息管理系统，忠诚的客户群，强大的融资能力。
- 竞争能力优势：产品开发周期短，强大的经销商网络，与供应商良好的伙伴关系，对市场环境变化的灵敏反应，市场份额的领导地位。

3.6.2 竞争劣势分析

竞争劣势（W）是指某种公司缺少或做得不好的东西，或指某种会使公司处于劣势的条件。

可能导致内部弱势的因素有：

- 缺乏具有竞争意义的技能技术。
- 缺乏有竞争力的有形资产、无形资产、人力资源、组织资产。
- 关键领域里的竞争能力正在丧失。

3.6.3　外部机会分析

市场机会是影响公司战略的重大因素。决策者应当确认每一个机会，评价每一个机会的成长和利润前景，选取那些可与公司财务和组织资源匹配、使公司获得最大竞争优势的最佳机会。

潜在的发展机会可能是：

- 客户群的扩大趋势或产品细分市场
- 技能技术向新产品新业务转移，为更大客户群服务
- 前向整合或后向整合
- 市场进入壁垒降低
- 获得并购竞争对手的能力
- 市场需求增长强劲，可快速扩张
- 出现向其他地理区域扩张，扩大市场份额的机会

3.6.4　外部威胁分析

在公司的外部环境中，总是存在某些对公司的赢利能力和市场地位构成威胁的因素。决策者应当及时确认危及公司未来利益的威胁，做出评价并采取相应的战略行动来抵消或减轻它们所产生的影响。

公司的外部威胁可能是：

- 出现将进入市场的强大的新竞争对手
- 替代品抢占公司销售额
- 主要产品市场增长率下降
- 汇率和外贸政策的不利变动
- 人口特征，社会消费方式的不利变动
- 客户或供应商的谈判能力提高
- 市场需求减少

- 容易受到经济萧条和业务周期的冲击

3.6.5 SWOT 核心问题

SWOT 分析方法的核心问题是：

- 我们的强项是什么？如何利用这些强项？
- 我们的弱项是什么？如何减少这些弱项的影响？
- 市场中有什么样的机会？如何把握这些机会？
- 有哪些威胁？如何有效地对付这些威胁？

第 4 章　需求分析

本章导读：需求分析是要决定“做什么，不做什么”。在用户导向的今天，用户需求分析变得越来越重要。在需求分析阶段，我们需要完成哪些工作？常用的需求分析工具有哪些？在本章中，这些问题会一一得到解答。

4.1 需求分析的重要性

软件研发的目地是为了满足人们的需求，因此不论是为客户做软件项目还是做软件产品，都要了解目标用户的真实需求，都要进行需求分析。

需求分析最难的地方是用户需求本身在不断改变！但同时需求分析又必须在软件开发启动之前完成。

软件研发的一个很重要的特点就是问题发现的越早，解决的代价就越低，假定在该阶段发现一个错误，解决它需要用一小时的时间，到设计、编程、测试和维护阶段解决，则要花 2.5 倍、5 倍、25 倍、100 倍的时间。

4.2 需求分析为什么困难

软件需求分析是整个软件开发过程中最令人烦恼的一个过程，“程序员往往并不怕软件所需技术本身有多难，就怕自己设计的软件不是用户想要的”。在需求分析阶段，分析人员往往也要面对以下挑战：需求本身在不停地改变；用户与开发人员很难交流；

分析人员或客户自身理解往往会发生错误。对于这些挑战，有什么样的对策？我们用一张表格来说明。

需求分析困难分析

	原　因	说　明	对　策
1	需求本身在不停地变	这是所有从事项目开发的软件企业心中永远的痛！用户的需求本身在不停地变化：企业的发展、业务变化，甚至领导更替都会导致需求的变化	（1）尽可能地分析清楚哪些是稳定的需求，哪些是易变的需求。以便在进行系统设计时，将软件的核心建筑在稳定的需求上，否则将会吃尽苦头 （2）在合同中一定要说清楚“做什么”和“不做什么” （3）“冻结需求”，即加快软件版本更新速度（如6个月），和用户达成共识：在此期限内用户新的需求只做记录、不做更新，等下一版本统一更新
2	用户与开发人员很难进行交流	大多数用户都不懂软件，所以想当然认为“冰箱的门从向右开改为向左开”是非常简单的事，殊不知对软件来说，整体架构都要变化，是多难的事	多和用户沟通，适当时候可以采用快速原型法，提供给用户 DEMO 版本，在此基础上，让用户验证需求
3	分析人员或客户理解有误	人与人之间的沟通障碍是非常普遍的，不同的人对同样的事会产生不同的理解，因此障碍就产生了	需求分析人员、开发人员，和用户一起参加需求分析，增加沟通交流机会，并共同就需求分析达成一致，并体现在合同和报告中

4.3 需求分析基本任务

需求分析的核心任务是了解以下方面用户的需求。

- 功能需求：明确所开发的软件必须具备什么样的功能。
- 性能需求：明确待开发的软件的技术性能指标。
- 环境需求：明确软件运行时所需要的软、硬件的要求。
- 用户界面需求：明确人机交互方式、输入输出数据格式。

4.4 需求分析的结果

- 完成“用户需求说明书”（可以参照本书模板篇提供的“用户需求说明书”模板）。
- 编写初步用户使用手册，着重反映被开发软件的用户功能界面和用户使用的具体要求，用户手册能强制分析人员从用户使用的观点考虑软件。
- 编写确认测试计划，作为今后确认和验收的依据。
- 修改完善软件开发计划。

4.5 需求分析阶段工作重点

需求分析分为三个阶段：需求调查、需求分析、需求评审。

需求分析阶段工作要点如下：

- 本项目中包含哪些用户？用户如何分类？
- 本项目中涉及哪些内容？
- 用户和内容的关系是什么？
- 用户的工作流程是什么？

4.6 UML 和 Rational Rose

在面向对象无孔不入的今天，利用面向对象的思想为软件系统建模，已经成为软件开发的主要工作，而传统的编码工作却“退居二线”了。一个系统的模型建得好，就为满足用户需求、保证系统的稳定性和质量、提高系统的扩展性打下了良好的基础。

设计一个软件的模型就好比是一幢大楼需要蓝图一样重要。好的模型能够：

- 鉴别需求和沟通信息。
- 着眼于系统的组件如何相互作用，而不是陷于具体的细节。
- 使你能够了解设计组件的相互关系。

- 通过使用一个共同的图形语言，改进跨团队的沟通。

UML（Unified Modeling Language，即统一建模语言）是一种定义良好、易于表达、功能强大且普遍适用的基于面向对象技术的标准建模语言。它融入了软件工程领域的新思想、新方法和新技术，支持从需求分析开始的软件开发全过程。

UML 通过建立各种类、类之间的关联、类/对象怎样相互配合实现系统的动态行为等成分（这些都称为模型元素）来组建整个模型，刻画客观世界。UML 提供了各种图形，比如 Use Case 图、类图、对象图、顺序图、协作图、状态图（共五类 10 种图）等，来把这些模型元素及其关系可视化，让人们可以清楚、容易地理解模型。我们可以从多个视角来考查模型，从而更加全面地了解模型，这样同一个模型元素可能会出现在多个图中，对应多个图形元素。

Rational Rose 是美国 Rational 公司的面向对象建模工具，利用这个工具，我们可以建立用 UML 描述的软件系统的模型，而且可以自动生成和维护 C++、Java、VB、Oracle 等语言和系统的代码。它完全采用了 UML 语言。2003 年随着 IBM 收购 Rational 公司，Rational Rose 也成为了 IBM 旗下的一个产品。

对于大型系统的开发，Rose 是一个很好的工具。它可以增强整个项目的可读性、可重用性，使得多个人的合作更加容易。但它的所有的优点是以多做文档、记录为代价的；对于小型系统或单个程序，它的优点是无法体现出来的。

4.7 Microsoft Visio

在需求分析和后续的系统设计阶段，经常要遇到画图的工作，比如用户的工作流程图、系统的执行流程图等。最常用的绘制图表的工具是 Visio。从 2000 版本后 Visio 被 Microsoft 收购，正式成为 Office 大家庭的一员。

Visio 提供了日常使用中的绝大多数框图的绘画功能（包括信息领域的各种原理图，设计图），同时提供了部分信息领域的实物图。Visio 的精华在于其使用方便，安装后的 Visio 既可以单独运行，也可以在 Word 中作为对象插入，与 Word 集成良好，其图生成后在没有安装 Visio 的 Word 仍然能够查看。

第 5 章　系统设计

本章导读：每个程序员都希望自己能够成为系统架构师。要想成为一名系统架构师，充分了解系统设计的相关知识、掌握系统设计的相关工具就显得非常重要。本章中，我们将就如何进行系统设计进行详细介绍。

5.1 系统设计工作流程图

系统设计分两个阶段：总体设计和详细设计。

（1）总体设计的主要任务是把需求分析得到的结果转换为软件结构和数据结构。设计软件结构的具体任务是：将一个复杂系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口及人机界面等。概要设计方法包括模块化方法、功能分解方法、面向数据流和面向数据结构的设计方法、面向对象的设计方法。

（2）详细设计的主要任务是设计每个模块的实现算法、所需的局部数据结构。详细设计的目标有两个：实现模块功能的算法要逻辑上正确，算法描述要简明易懂。

系统设计工作流程图如图 5-1 所示。

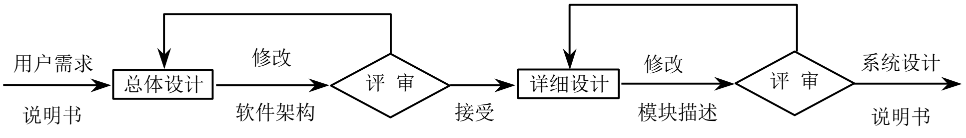


图 5-1　系统设计工作流程

5.2 面向对象的设计方法

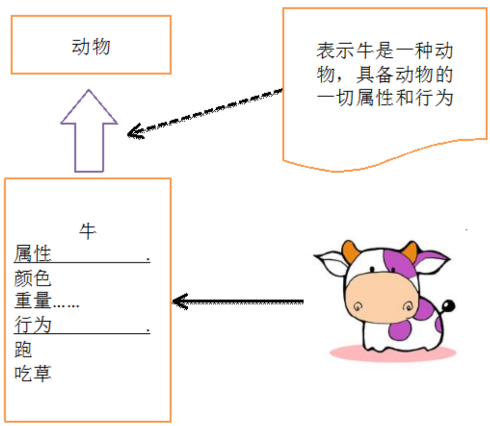
20 世纪 80 年代以来，OO（Object Oriented）技术在计算机软件开发技术研究和应用领域中十分活跃，它的优越性几乎在一夜之间就被人们所接受。许多专家和学者预言：正如 70 年代以来的结构化方法对计算机技术的应用所产生的巨大影响和促进一样，80 年代以来的 OO 技术将从人们的认识论、方法论、系统分析与设计编程风格、测试方法与策略等领域强烈地影响和推动一系列计算机高新技术的发展。

面向对象是一种方法，一种思想，同时又是一种技术。它力求更客观自然地描述现实世界，使分析、设计和实现系统的方法同认识客观世界的过程尽可能一致。

实践证明，任何现实问题领域不外乎是由一些基本原始事物组成的，这些事物之间存在着一定的联系。用计算机系统求解的问题都是现实世界的具体问题，根据求解问题的目的将现实世界问题作相应限定，就获得求解问题的问题空间，经抽象规范化处理就获得了计算机求解问题空间，再经计算机求解并获得问题的解。

OO 思想的基本原理是：按照问题领域的基本事物实现自然分割，按人们通常的思维模式建立问题领域的模型，设计尽可能直接自然表现问题求解的软件系统。对象表现事物，用消息传递建立事物间的联系。

客观世界是由许多不同种类的对象构成的，每一个对象都有自己的运动规律和内部状态，不同对象之间相互联系、相互作用。“面向对象技术”是一种从组织上模拟客观世界的方法，从组成客观世界的对象着眼，通过抽象，将对象映射到计算机系统，又通过模拟对象之间的相互作用、互相联系来模拟现实客观世界，描述客观世界的运动规律。



用面向对象的方法来认识牛

5.2.1 面向对象的概念

面向对象（OO）是认识事物的一种方法，是一种以对象为中心的思维方式。这里介绍它的几个主要概念：对象、类、封装、继承、多态性、消息、结构与连接。

1．对象

世界上所有的事物都可以称为对象。对象可以是有形的，例如：一台电视机等。也可以是无形的，例如：账户、一项记录等。对象具有静态特征和动态特征。

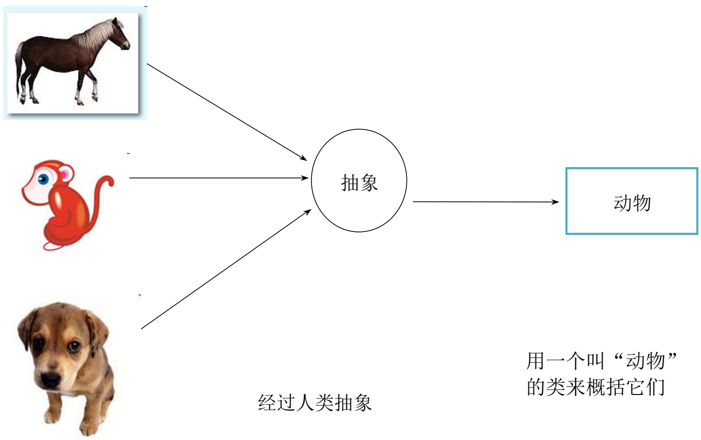
一个对象就是一个独立存在的客观事物，它由一组属性和对属性进行操作的一组操作构成。

属性和操作是对象的两大要素。属性是对象静态特征的描述，操作是对象动态特征的描述。对象名也称为对象标识，如电视的属性有：品牌、尺寸、重量等。操作有：收视、选台、音量调节等。

2．类

根据抽象的原则对客观事物进行归纳和划分，只关注与当前目标相关的特征，把具有相同特征的事物归为一个类。它是一个抽象的概念。

类是具有相同属性和相同操作（服务）的对象的集合。它包括属性和操作（注：类的服务和操作只是叫法上的区别）。



客观世界中的每一匹马都属于动物类，其中的一匹马就是动物类的一个实例，即一个动物对象。

3．封装

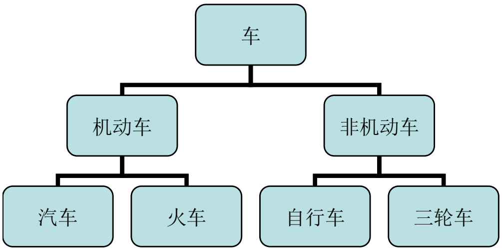
封装是指按照信息屏蔽的原则，把对象的属性和操作结合在一起，构成一个独立的对象。

外部对象不能直接操作对象的属性，只能使用对象提供的服务。例如，我们不用关心电视机的内部工作原理，电视机提供了选台、调节音量等功能让我们使用。

4．继承

继承表达了对象的一般与特殊的关系。特殊类的对象具有一般类的全部属性和服务。

机动车具有车的全部属性和服务，同样汽车具有机动车的全部属性和服务。



一般和特殊是相对而言的，在车和机动车之间，车是一般类（基类、超类、父类），机动车是特殊类（子类）；在机动车和汽车之间，机动车是一般类，汽车是特殊类。

继承具有传递性，如汽车具有车的全部属性和服务。

5．消息

向某个对象发出的服务请求称作消息。对象提供的服务规定的消息格式称作消息协议。

消息包括：被请求的对象标识、被请求的服务标识、输入信息和应答信息。

6．结构与连接

一个系统一般由很多对象组成，对象之间并不是互相孤立的，而是存在着各种各样的关系。包括：部分/整体、一般/特殊、实例连接、消息连接。

- 部分/整体

对象之间存在的部分与整体的结构关系。如 CPU 是计算机的一个组成部分，计算机包括主板、显卡、声卡、内存、硬盘、显示器等。

部分/整体关系中有两种方式：组合和聚合。

计算机就是一个组合的例子，一个部分对象（CPU）只能属于一个唯一的整体对象

（计算机）。组合关系中部分和整体的关系很紧密。聚合关系中则比较松散，一个部分对象可以属于几个整体对象。

- 一般/特殊

对象之间存在着一般和特殊的结构关系，也就是说它们存在继承关系。很多时候也称作泛化和特化关系。

- 实例连接

实例连接表现了对象之间的静态联系，它通过对象的属性来表现出对象之间的依赖关系。对象之间的实例连接称作链接，对象类之间的实例连接称作关联。

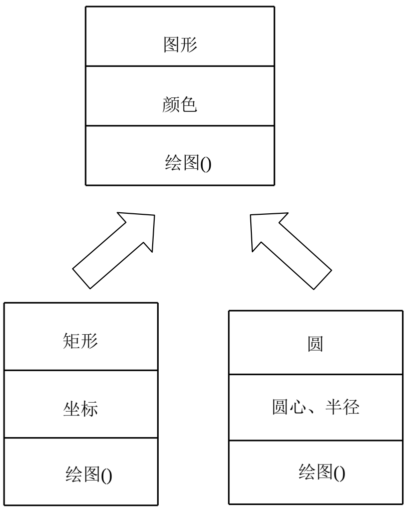
- 消息连接

消息连接表现了对象之间的动态联系，它表现了这样一种联系：一个对象发送消息请求另一个对象的服务，接收消息的对象响应消息，执行相应的服务。

7. 多态性

多态性是指一般类中定义的属性和服务，在特殊类中不改变其名字，但通过各自不同的实现后，可以具有不同的数据类型或具有不同的行为。

如一个绘图系统中类的多态性：



当向图形对象发送消息进行绘图服务请求后，图形对象会自动判断自己的所属类然后执行相应的绘图服务。

5.2.2 面向对象设计方法目标

在系统设计时，设计人员如果能够达到以下“六化”，即模块化、角色化、流程化、规范化、简单化、个性化，那最后的设计结果将会是非常令人满意的。我们用表格总结如下。

面向对象设计目标

	特　点	说　明
1	模块化	把整个系统划分成几个相互关联的模板
2	角色化	需要分别从不同用户的角度出发去考虑其工作流程
3	流程化	需要把用户使用、工作的过程流程化，并在系统中体现
4	规范化	需要符合一定的标准
5	简单化	系统的使用要用户友好，尽量简洁
6	个性化	对于不同的用户要能够体现其个性化的东西

5.3 三层应用程序模型

三层应用程序模型是指在进行系统设计时，整个系统分成三个层次：用户界面层（UI层）、业务对象层（逻辑层）、数据层。

用户对象层主要注重于软件界面的实现；业务对象层关注于用户的业务规则和逻辑结构；数据层关注于数据库的实现，如图5-2所示。

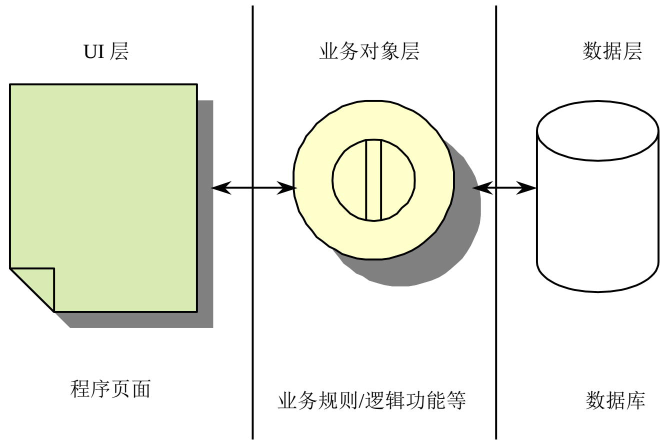


图5-2　三层应用程序模型

为了更好地理解三层应用程序模型，我们做一个比喻：

- 用户界面层（UI）：如同舞台上的演员，吸引观众的注意力，给观众体验等。
- 逻辑层：提供指示和提示的人，包括管弦乐团、舞台管理人员、导演等。这些人与舞台上的演员打交道，但观众看不到。他们指挥演员，并给演员提供材料。
- 数据层：由负责布景和素材的人员组成，包括编剧、舞美、布景师等。这些人共同提供了戏剧素材，观众只能看到他们的作品。

电子商务中：UI 为表单、购物车、图形等；业务层为控制定价、运输费等逻辑组成；数据层由数据库中的存货清单组成。

在很多情况下，我们进行系统设计都会遵循三层应用程序模型。

5.4 函数、对象和组件

软件开发中，最重要的一个目标就是要追求软件的重复使用。

5.4.1 函数和过程

在开始面向过程设计的年代，人们发明了“过程”和“函数”。把经常使用的代码或者有一定功能的代码放到一个过程或函数中，就可以在程序中的不同地方重复使用这些代码了，这不仅避免了每次都重复输入这些代码，同时也降低了错误率，函数出错只需要修改函数内部就可以了。

5.4.2 类和对象

后来，整个软件开发进入了面向对象的设计年代。这时，由于面向对象技术的蓬勃发展，很多的事务都可以用“类”来描述了。在程序的不同地方，同一个类，只需要实例化一下（称为对象），就可以使用预先定义好的功能了。这使得程序的复用性进一步增强。

5.4.3 组件

接下来是组件技术的出现。

组件是什么？

- 组件是可以在不同的应用程序中使用的对象集合。
- 组件是功能更完备的单元，带有与它的内部体系结构相对独立的外部接口。

组件的优点：

- 单个组件比一个功能完备的应用程序要简单得多。它仅仅是一组预先定义好的功能。
- 组件功能是相对独立的，如果接口相同，则可以拿另外一个组件来代替原来的组件，可以做到无缝升级。
- 组件是可以重复使用的。
- 组件可以减少代码量，使得代码更容易维护。

我们可以把组件理解为软件编程中的积木。

微软是应用组件技术的典范，Windows 中有很多的 DLL（动态链接库）就是一些组件。如.NET 类库就是如此，通过这些类库，程序开发者可以很容易地使用微软已经预先设计好的功能，从而使得软件的编码变得更加容易。

另外，Com（Compoent object model）、EJB、.NET 等技术的出现，也是组件技术的一个发展。通过组件技术，软件开发的复用性得到了空前的发展。

整个软件开发的过程中，要充分地考虑代码复用性问题。在系统设计和编码阶段，应该考虑是否能用函数（过程）、对象、组件的技术来充分提高代码的复用性。

5.5 系统设计内容

在系统设计阶段，我们认为主要包括以下几个方面的设计工作，即数据库设计、界面设计、模块设计、数据结构和算法设计等。

系统设计过程示意图如图 5-3 所示。

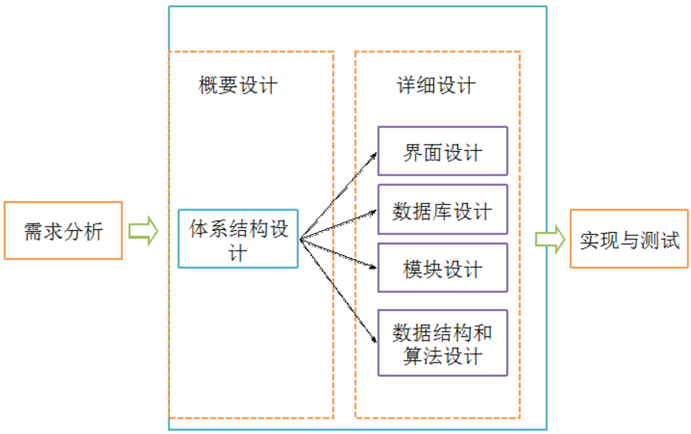
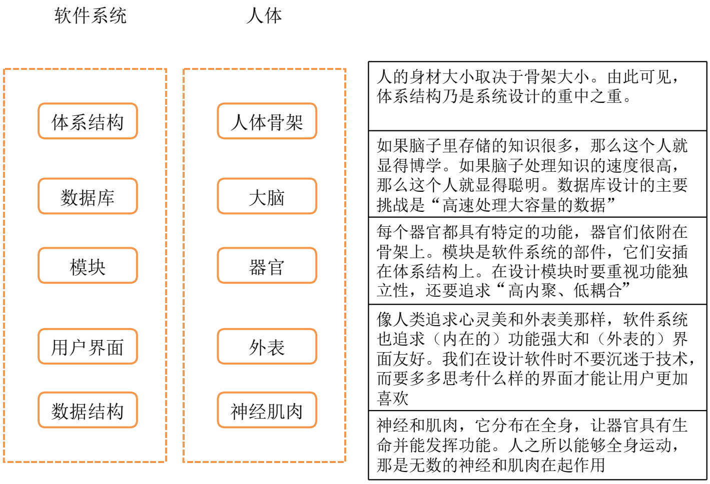


图 5-3 系统设计过程示意图

我们以人体来比喻系统分析：



5.5.1　数据库设计

数据库是用来存储和处理数据的。数据库设计的主要目标是：

- 设计数据库的表（数据就存在表里面），表的结构就是数据的存储结构。
- 对这些表中的数据进行操作，常见操作如查询、插入、修改、删除等。

1．ER 图

程序设计的本质是要把用户的工作流程反映为程序的执行流程，即把现实世界转化为计算机能理解的模型。数据库设计的本质是要把现实世界的数据抽象为计算机内部表示的数据特征模型。

概念数据模型也称信息模型，它以实体一联系（Entity-RelationShip，简称 ER）理论为基础，并对这一理论进行了扩充。它从用户的观点出发对信息进行建模，主要用于数据库的概念级设计。

数据库设计分为两个步骤：

（1）先将现实世界中的客观对象抽象为实体（Entity）和联系（Relationship）。这种模型在 PowerDesigner 中被称为概念数据模型（CDM）。

（2）然后再将 CDM 转换为计算机上某个数据库系统（DBMS）所支持的数据模型。在 PowerDesigner 中被称为物理数据模型（PDM）。

ER 图所使用的符号包括如下：

符　　号	名　　称	意　　义
▭	实体	实体是指数据库内所需要储存数据的人、事、地、物等，是一个变量的概念，代表某种东西，是一个关系型
⬭	属性	实体的特性，描写实体的项目，比如说，学生是一个实体，学生的属性有学号、姓名、性别、班级、地址、电话等，是关系型中的字段
◇	关系	实体和实体之间的关系，有一对一、一对多和多对多，当多对多的关系时，关系也是一个关系型

ER 图的画法如下：

- 实体型（Entity）：用矩形表示，矩形框内写明实体名。
- 属性（Attribute）：用椭圆形表示，并用无向边将其与相应的实体连接起来；比如学生的姓名、学号、性别都是属性。
- 联系（Relationship）：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型（1:1，1:*n* 或 *m*:*n*）就是指存在的三种关系（一对一，一对多，多对多）。比如夫妻关系就是 1:1 关系，父子关系就是可能是 1:*n*，学生和课程的关系就是多对多关系。

图 5-4 给出的是一个学校管理系统的数据库 ER 图。

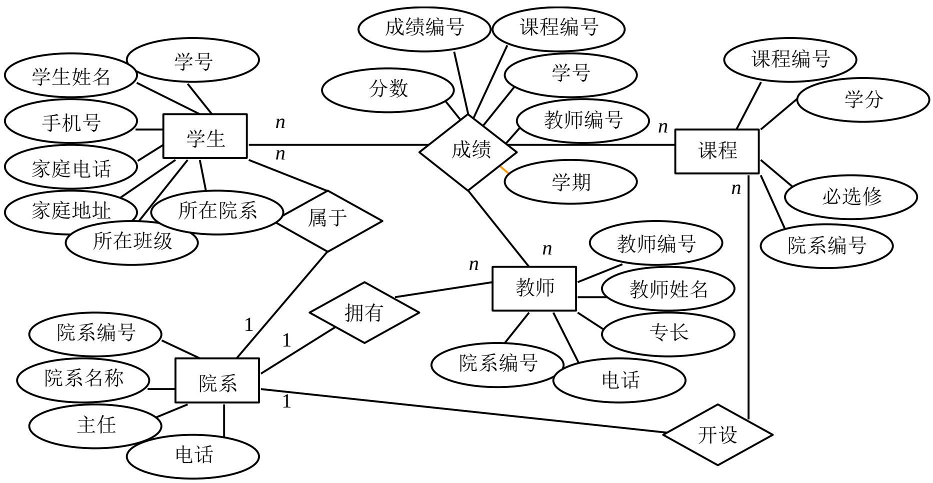


图 5-4　学校管理系统数据库 ER 图

2．引例：商品信息表的设计

假如你们公司正在为一家企业设计一套网上商城系统，该企业有数千种商品，不过目前正处于尝试阶段，仅打算在网上销售数十种方便运输的商品。剩余的商品在以后陆续推出。

你代表你们公司负责这个项目的设计开发，经过和用户的接洽和调研，你开始着手进行数据库的设计。

很显然，根据对商品属性的分析，每种出售的商品都会有相同的属性，如商品编号、商品名称、商品所属类别、相关信息、供货厂商、内含件数、库存、进货价、销售价、优惠价。你很快就设计出 3 个表：商品类型表（ProductsType）、供货厂商表（ProductsProvider）和商品信息表（ProductsInfo）。

商品类型表（ProductsType）：

名　称	类　型	约束条件	说　明
type_id	int	无重复	类别标识，主键
type_name	char(50)	不允许为空	类型名称，不允许重复
type_father	int	不允许为空	该类别的父类别标识，如果是顶节点的话设定为某个唯一值
type_layer	char(6)	限定 3 层，初始值为 000000	类别的先序遍历，主要为减少检索数据库的次数

供货厂商表（ProductsProvider）：

名　称	类　型	约束条件	说　明
provider_id	int	无重复	供货商标识，主键
provider_name	char(100)	不允许为空	供货商名称

商品信息表（ProductsInfo）：

名　称	类　型	约束条件	说　明
products_id	int	无重复	商品标识，主键
products_name	char(100)	不允许为空	商品名称
products_type	int	不允许为空	商品类型标识和 ProductsType.type_id 关联
products_info	char(200)	允许为空	相关信息
provider	int	不允许为空	供货厂商标识和 ProductsProvider.provider_id 关联
setnum	int	初始值为 1	内含件数，默认为 1
stock	int	初始值为 0	库存，默认为 0
buy_price	money	不允许为空	进货价

续表

名　称	类　型	约束条件	说　明
sell_price	money	不允许为空	销售价
discount	money	不允许为空	优惠价

设计完成后，你拿着这 3 张表给老板检查，老板希望能够再添加一个商品图片的字段，不过只有一部分商品有图片。于是，你在商品信息表（ProductsInfo）中增加了一个 haspic 的 BOOL 型字段，然后再建了一张新表——商品图片表（ProductsPic）。

商品图片表（ProductsPic）：

名　称	类　型	约束条件	说　明
pic_id	int	无重复	商品图片标识，主键
products_id	int	不允许为空	所属商品标识和 ProductsInfo. products_id 关联
pic_address	char(200)	不允许为空	图片存放路径

一段时间后，用户打算在这套平台上推出新的商品销售，其中，某类商品全部都需添加“长度”的属性。

没有办法，你只好按照添加商品图片表的老方法，在商品信息表（ProductsInfo）中增加了一个 haslength 的 BOOL 型字段，又建了一张新表——商品长度表（ProductLength）。

商品长度表（ProductLength）：

名　称	类　型	约束条件	说　明
length_id	int	无重复	商品图片标识，主键
products_id	int	不允许为空	所属商品标识和 ProductsInfo. products_id 关联
length	char(20)	不允许为空	商品长度说明

过了一段时间后，用户又有新的商品上架，又要增加商品“宽度”、“高度”……正如我们前面所述，用户的需求在不停地变化。每一次修改数据库都是大量的工作。你几乎要崩溃了！

那应该如何办？

去掉商品信息表（ProductsInfo）中的 haspic 字段，添加商品额外属性表（ProductsExProperty）和商品额外信息表（ProductsExInfo）两张表来完成添加新属性的功能。

商品额外属性表（ProductsExProperty）：

名　　称	类　　型	约 束 条 件	说　　明
ex_pid	int	无重复	商品额外属性标识，主键
p_name	char(20)	不允许为空	额外属性名称

商品额外信息表（ProductsExInfo）：

名　　称	类　　型	约 束 条 件	说　　明
ex_iid	int	无重复	商品额外信息标识，主键
products_id	int	不允许为空	所属商品标识和 ProductsInfo .products_id 关联
property_id	int	不允许为空	商品额外属性标识和 ProductsExProperty.ex_pid 关联
property_value	char(200)	不允许为空	商品额外属性值

终于搞定！无论再有多少新的商品属性都不害怕了！

从以上的例子可以看出，数据库的设计有多么重要！不恰当的数据库设计会对以后的扩展工作带来很多麻烦。

3．数据库命名规范

一个项目组内所有设计数据库的程序员应当共同遵守统一的“数据库命名规范”。在本书的附录 B 中，我们提供了一个“数据库命名规范”实例，供大家参考。

4．数据库设计范式

关系数据库设计之时是要遵守一定的规则的。尤其是数据库设计范式，接下来简单介绍 1NF（第一范式），2NF（第二范式），3NF（第三范式）和 BCNF，另外还有第四范式和第五范式。在你设计数据库时，若能符合这几个范式，你就是数据库设计的高手。

（1）第一范式（1NF）

数据库表中的字段都是单一属性的，不可再分。这个单一属性由基本类型构成，包括整型、实数、字符型、逻辑型、日期型等。

例如，电话的字段就可能不符合第一正规化，因为每个人可能有一个以上的电话需要记录，这时最简单的做法就是将字段增加，不要用电话当字段名称，用家里电话、手机、办公室电话、分机号码、宿舍电话等。

学　　生			⇨	学　　生			
学号	姓名	电话		学号	姓名	手机	电话（H）
901026001	刘德华	4631070,13412121212		901026001	刘德华	13412121212	4631070

续表

学　　生			⇨	学　　生			
学号	姓名	电话		学号	姓名	手机	电话（H）
901026002	周润发	4251245,13666546666		901026002	周润发	13666546666	4251245
901026003	成龙	6841245,13786424221		901026003	成龙	13786424221	26841245
…	..	…		…	..	..	..

（2）第二范式（2NF）

规则是符合第一范式，而且没有部分主键功能决定其他属性的现象，也就是主键之外的其他属性都完全的功能相依于主键。

假定选课关系表为 SelectCourse（学号，姓名，年龄，课程名称，成绩，学分），关键字为组合关键字（学号，课程名称），因为存在如下决定关系：（学号，课程名称）→（姓名，年龄，成绩，学分）。

这个数据库表不满足第二范式，因为存在如下决定关系：

- （课程名称）→（学分），即知道课程，就可以知道该课程的学分。
- （学号）→（姓名，年龄），即知道了学号，就可以知道该学生的姓名和年龄。

① 数据冗余

同一门课程由 n 个学生选修，“学分”就重复 $n-1$ 次；同一个学生选修了 m 门课程，姓名和年龄就重复了 $m-1$ 次。

② 更新异常

若调整了某门课程的学分，数据表中所有行的“学分”值都要更新，否则会出现同一门课程学分不同的情况。

③ 插入异常

假设要开设一门新的课程，暂时还没有人选修。这样，由于还没有“学号”关键字，课程名称和学分也无法记录入数据库。

④ 删除异常

假设一批学生已经完成课程的选修，这些选修记录就应该从数据库表中删除。但是，与此同时，课程名称和学分信息也被删除了。很显然，这也会导致插入异常。

把选课关系表 SelectCourse 改为如下三个表。

- 学生：Student（学号，姓名，年龄）
- 课程：Course（课程名称，学分）
- 选课关系：SelectCourse（学号，课程名称，成绩）

这样的数据库表是符合第二范式的，消除了数据冗余、更新异常、插入异常和删除异常。

（3）第三范式（3NF）

在第二范式的基础上，数据表中如果不存在非关键字段对任一候选关键字段的传递函数依赖则符合第三范式。

所谓传递函数依赖，指的是如果存在“A→B→C”的决定关系，则 C 传递函数依赖于 A。因此，满足第三范式的数据库表应该不存在如下依赖关系：关键字段→非关键字段 x→非关键字段 y。

假定学生关系表为 Student（学号，姓名，年龄，所在学院，学院地点，学院电话），关键字为单一关键字“学号”，因为存在如下决定关系：

- （学号）→（姓名，年龄，所在学院，学院地点，学院电话）

这个数据库是符合 2NF 的，但是不符合 3NF，因为存在如下决定关系：

（学号）→（所在学院）→（学院地点，学院电话）

把学生关系表分为如下两个表。

- 学生：（学号，姓名，年龄，所在学院）；
- 学院：（学院，地点，电话）。

（4）BCNF 范式

BCNF 范式（Boyce/Codd Normal Form），是由 R.F.Boycy 和 E.F. Codd 共同提出的，可以算成是第三正则化的补充，规则是符合第三正则化原则，并且没有非主键属性可以功能性决定部分主键的现象。

假设有一个表 R，其中的属性有 A，B，C，D，E，以 A 和 B 为复合主键，R={A，B，C，D，E}，如果存在有非主键属性，比如说 C 可以功能性决定 B，C→B，而 B 是主键的一部分，这时第三正则化是没有办法分辨出来这种错误的，所以有 BCNF 正则化规则来把关，同样地，BCNF 正则化的方法也是将原来的表拆开，成立一个新的关联表 R1 来装 C→B，R1={C，B}，但原来的表 R 还是以（A，B）为复合主键，以 B 为外键

关联到新的表去，以保留原有的信息。

R={A,B,D,E}，R1={C,B}，R.B=R1.B

小结

在表的物理设计阶段，设计人员应当按照第三范式设计表结构（即规范化处理）。这样做的好处：表中没有冗余数据，表结构很清晰，将来修改或者扩充非常方便。但是按第三范式设计也存在一些缺点：产生了许多表，每个表又相对较少的列，并且这些列必须使用“主键/外键”关联起来，因此某个查询操作可能会产生复杂的表链接，导致性能降低。

反规范化处理是指对第三范式的表进行修改，通过合并一些表，或者在表中创建冗余的列，从而减少表链接操作代价，达到提高性能的目的。要注意的是，反规范化处理存在很大的负面影响：管理冗余数据很麻烦，如果冗余数据不同步的话，那么会发生数据错误这种严重的问题。

对表进行第三范式的规范化处理是第一重要的，而反规范化处理则需要谨慎考虑、不宜过多使用。“规范化处理”，以及“反规范化处理”不是自相矛盾之举，而是性能优化的策略。

5．正则化技巧

正则化的过程是检查数据库表格的设计，并且加以调整，让表格内的字段的建置能符合设计的原则，在新增、更新、删除记录数据时不至于发生错误的过程。

为了使得讲解更加清楚，我们举例来说明。

这里假定要建立一个用户信息的表格，其中要存储用户的名字、公司、公司地址和一些个人的收藏夹或URL。在开始时，你可能定义一个如下的表格结构。

（1）零状态形式

Users表：

Name	Company	CompanyAddress	URL1	URL2
张三	IBM	北京海淀	abc.com	Xyz.com
李四	MS	北京朝阳	abc.com	Xyz.com

但上面的表格中存在以下问题：如果我们要给每个用户添加一个新的URL地址，就需要更改表结构，所以以上的表格中存在“扩展性问题”。

（2）解决“扩展性问题”

我们把 URL1 和 URL2 两个字段删除，建立一个新的字段 URL。每插入一个新的 URL 就在数据库中添加一条新的记录，这样就解决了数据表的“扩展性问题”。

如下表所示：

UserId	Name	Company	CompanyAddress	URL
1	张三	IBM	北京海淀	abc.com
1	张三	IBM	北京海淀	Xyz.com
2	李四	MS	北京朝阳	abc.com
2	李四	MS	北京朝阳	Xyz.com

以上的表结构，基本解决了“扩展性问题”，但又发现新的问题：“数据重复问题”。很显然，对于每个用户要添加新的 URL 时，需要在数据库中添加新的记录，数据重复性很严重。

（3）解决“数据重复问题”

为了解决数据重复问题，我们把以上的表格分成两张表。

Users 表：

UserId	Name	Company	CompanyAddress
1	张三	IBM	北京海淀
2	李四	MS	北京朝阳

URLs 表：

URLID	RelUserId	URL
1	1	abc.com
2	1	Xyz.com
3	2	abc.com
4	2	Xyz.com

通过使用以上两张表，我们解决了上述提到的“数据重复问题”。但经过我们仔细研究发现，在 Users 表中，不同的用户可能在同一个公司中，而 CompanyAddress 字段显然和某个用户是相对独立的，对于处于同一公司的用户，上述两个字段还会出现“数据重复问题”。

（4）进一步解决“数据重复问题”

为了解决上述问题，我们进一步把 Users 表拆分成两张表：Companies 和 Users。

Companies 表：

CompId	Company	CompanyAddress
1	IBM	北京海淀
2	MS	北京朝阳

Users 表：

UserId	Name	RelCompID
1	张三	1
2	李四	2

URLs 表：

UrlID	RelUserId	URL
1	1	abc.com
2	1	Xyz.com
3	2	abc.com
4	2	Xyz.com

对于以上的 URLs 表，我们再做进一步分析，发现像“abc.com”和“Xyz.com”这样的数据都出现了两次，还存在“数据冗余问题”。

（5）解决“数据冗余问题”

为了解决“数据冗余问题，我们把 URLs 表格再进行拆分。

URLs：

URLID	URL
1	abc.com
2	Xyz.com
3	abc.com
4	Xyz.com

URL_relations：

relationID	relationURLId	relatedUseID
1	1	1
2	1	2
3	2	1
4	2	2

通过以上 5 步操作，我们建好了 Companies 表、Users 表、URLs 表和 URL_relations 表，这些表基本能够满足我们的需要了。

小 结

通过以上的这个具体实例分析，大家了解了数据库正则化的一些技巧和设计数据库结构时我们应该注意的一些问题。但是要注意，随着正则化的程度越深，关联表会越多，需要越多的关联来存取，如此一来，对数据库执行上的效率而言，会比较有问题，所以在实际操作上并不会将所有的正则化规则全部应用，通常会做到 BCNF。

6．PowerDisigner 简介

Power Designer 是 Sybase 公司的 CASE 工具集，使用它可以方便地对管理信息系统进行分析设计，它几乎包括了数据库模型设计的全过程。利用 Power Designer 可以制作数据流程图、概念数据模型、物理数据模型，可以生成多种客户端开发工具的应用程序，还可以为数据仓库制作结构模型，也能对团队设计模型进行控制。

运用 PowerDesigner 进行数据库设计，不但使人直观地理解模型，而且充分运用数据库的技术，优化数据库的设计。PowerDesigner 支持 Sybase、Oracle、Informix、SQL Server 等多种数据库系统，在应用系统做数据库迁移时不必维护多个数据库脚本。它可以与许多流行的程序设计软件，例如：PowerBuilder、Delphi、VB 等相配合使用来缩短开发时间和使系统设计更优化。

PowerDesigner 包含 6 个紧密集成的模块，允许个人和开发组的成员以合算的方式更好地满足他们的需要。这 6 个模块是

- PowerDesigner ProcessAnalyst：用于数据发现。
- PowerDesigner DataArchitect：用于双层，交互式的数据库设计和构造。
- PowerDesigner AppModeler：用于物理建模和应用对象及数据敏感组件的生成。
- PowerDesigner MetaWorks：用于高级的团队开发，信息的共享和模型的管理。
- PowerDesigner WarehouseArchitect：用于数据仓库的设计和实现。
- PowerDesigner Viewer：用于以只读的、图形化方式访问整个企业的模型信息。

5.5.2 界面设计

软件的界面如同人的脸一样，软件界面的好坏决定用户对软件的第一印象。设计良

好的界面能够引导用户自己完成相应的操作，起到向导的作用。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉。相反，由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。一些专家指出：对于用户，人机界面就是系统本身。这充分说明了软件界面设计的重要性。

常见的软件界面类型包括

- 对话框界面：如附件中的“计算器”。
- 单文档界面：如附件中的“记事本”。
- 多文档界面：如 Office。
- 浏览器界面：如 Outlook Express。
- 其他类型。

现在大多数的网络应用程序都是浏览器界面，即 B/S（Browser/Server）结构的。

接下来我们详细介绍用户界面设计的一些原则。

1．用户界面设计原则

用户界面的设计包括以下一些原则。

（1）用户界面适合于软件的功能

“用户界面适合于软件的功能”是基本的设计原则，它提醒设计者不要片面地追求界面外观漂亮而导致华而不实。界面的合适性既提倡外美内秀，又强调恰如其分。

例如，办公系统的软件界面要设计得简洁、大方，而娱乐软件，如网络游戏的软件界面设计就要设计得活泼、生动。

（2）用户友好原则

在软件设计中有一个非常重要的原则，即“用户友好原则”。在软件用户界面设计中，也要遵循这个原则。它包括以下一些内容：

- 控件位置得当（一般左上角放重要的信息；工作将要完成时才用到的元素如“确定”、“提交”等按钮放在右下角）。
- 控件组合适当（例如，可以灵活使用 GroupBox 等控件进行界面组合）。
- 协调。

- 颜色和字体的恰当选择（一个界面上，一般字体在三种之内，颜色在 16 色之内）。
- 符合习惯（如我们在规范篇中提到的界面设计规范）。
- 易于使用（如快捷键、Tab 次序、右键等的设置）。

（3）一致性原则

一个应用程序中的多个界面应该保持风格的一致，这是最基本的要求。一致的软件界面不会增加用户的负担，让用户始终用同一种方式思考与操作。最忌讳的是每换一个屏幕用户就要换一套操作命令与操作方法。

Windows 下的应用软件之所以备受青睐，与其界面的一致性不无关系。例如，以问号图标表示帮助，以磁盘图标表示存盘，以打印机图标表示打印，等等。

对于用户界面而言，风格一致的最大好处就是能够减小用户的记忆量、减小出错几率，并且迅速积累操作经验。所以熟悉 Word 软件的人基本上不用翻阅手册就能使用 PowerPoint 软件，可谓“无师自通”。

（4）及时反馈信息

软件界面应该能够告诉用户软件运行的进度。特别是在需要较长时间的等待时，必须让用户了解工作进展情况，如已经完成了百分之几等。切不可让用户面对一个没有反应的屏幕，以致怀疑是否出现了死机现象。

（5）出错处理

用户在使用软件的过程中，不可避免地会出现一些错误的操作。在设计用户界面时必须考虑出错处理，目的是让用户不必为避免犯错误而提心吊胆、小心翼翼地操作。

常见的错误处理方式有：

- 提供对输入数据进行校验的功能。当用户输入错误的数据时，及时提醒用户改正数据。
- 对于在某些情况下不应该使用的菜单项和命令按钮，将其“失效”（屏蔽）可以有效防止该项功能被错误地使用。例如：对于某些管理软件，不同的用户有不同的操作权限。如果低权限的用户登录到系统，那些只有高级权限用户才能使用的功能应当被屏蔽（如变成“灰色”不可操作）。
- 提供 Undo 功能，用以撤销不期望的操作。

- 执行破坏性的操作之前，应当获得用户的确认。例如，用户删除一个文件时，应当弹出对话框："真的要删除该文件吗？"当用户确认后才真正删除文件。

（6）帮助和提示原则

要对用户的操作命令做出反应，帮助用户处理问题。系统要设计有恢复出错现场的能力，在系统内部处理工作要有提示，尽量把主动权让给用户。

（7）信息最小量原则

软件界面设计要尽量减轻用户记忆负担，采用有助于记忆的设计方案。

2. 用户界面设计注意事项

界面设计注意事项

	项	说　明
1	启动界面设计（SPLASH）	考虑不同的操作系统，对选用的色彩不宜超过 256 色。软件启动封面大小多为主流显示器分辨率的 1/6 大
2	安装界面设计	包括安装进度过程设计、安装选项设计，应该尽量做到用户友好，把必要的安装项都放入默认安装项
3	软件架构设计	软件框架设计应该简洁明快，尽量少用无谓的装饰，应该考虑节省屏幕空间，各种分辨率的大小，缩放时的状态和原则，并且为将来设计的按钮、菜单、标签、滚动条及状态栏预留位置。设计中将整体色彩组合进行合理搭配，将软件商标放在显著位置，主菜单应放在左边或上边，滚动条放在右边，状态栏放在下边，以符合视觉流程和用户使用心理
4	按钮设计	一般有三种或三种以上的状态效果：默认状态；鼠标移上去的状态；单击时状态；不能单击时状态。按钮应具备简洁的图示效果，应能够让使用者产生功能关联反应，群组内按钮应该风格统一，功能差异大的按钮应该有所区别
5	面板设计	软件面板功能区分割清晰，且具有缩放功能，与整个软件界面风格匹配，尽量做到面积较小，节省空间
6	菜单设计	一般有选中状态和默认状态，右边应为快捷键，左边应为名称，如果有下级菜单应该有提示（如箭头等），不同功能区间应该分开
7	图标设计	色彩一般不宜超过 64 色，大小为 16×16、32×32、64×64 等几种，图标设计是方寸艺术，它需要在小面积内表现出软件的功能的含义，因此多数设计人员设计图标时使用较少的颜色，利用人们对色彩和网点的空间混合效果，做出了许多精致的图标
8	滚动条设计及状态栏设计	主要是为了对区域性空间的固定大小中内容量的变换进行设计，应该有上、下箭头，滚动标，还有翻页标。状态栏设计是为了对软件当前状态的显示和提示

3. 用户界面设计规范

在“规范篇”中我们提供了“用户界面设计规范”模板，供大家在实际项目开发中使用。

5.5.3 模块设计

1.“模块”与“模块化”

模块泛指软件系统的功能部件。在软件的体系结构设计完成之际，我们就已经确定了所有模块的功能，并且把模块安放在体系结构的恰当位置上。

每个模块都具有特定的、明确的功能（否则不能成为模块）。人们在设计模块时应当尽量使模块的功能独立，因为功能独立的模块可以降低开发、测试、维护的代价。但是功能独立并不意味着模块是绝对孤立的。所有的模块应当能够被集成为一个系统，所以模块之间必定要交流信息、相互配合。

模块类似于我们玩“搭积木”游戏：我们用积木搭成一个房子，每块积木可以被看成是一个模块，它们都是整个系统（房子）的一个部分，并且有特定的功能；各个积木之间还有豁口，可以组合，如同各个模块之间的接口，同时各个积木又是相互独立的。

“模块化“（Modularization）是指，将系统分解为一系列功能模块，然后逐一实现这些模块，最后把所有的模块集成为原来的系统。这样做的好处是能够大大降低系统的开发难度。

再比如组装计算机，我们去电子商场买到了组装计算机所用的各个模块（主板、内存、CPU、显卡、声卡、机箱、键盘、鼠标、显示器），然后把它们组装在了一起。把计算机分成一个个的模块，使得类似于我们这样的普通用户也可以对计算机进行简单的维修（最起码可以换某一个部件）。这种好处是显而易见的。

是否将系统分解得越细，得到的功能模块越多越好呢？不是的。虽然这样做可以使实现模块的代价更低，但是把功能模块集成为原来系统的代价却增大了很多，得不偿失，所以一个系统的模块数量不能过多也不能过少。那么多少算是恰如其分呢？不知道，要靠设计师的判断。

2. 模块设计原则

（1）信息隐藏

- 为了尽量避免某个模块的行为干扰同一系统中的其他模块，在设计模块时就要注意

信息隐藏。应该让模块仅仅公开必须要让外界知道的东西，而隐藏其他一切内容。

- 接口设计是模块设计的核心工作之一，体现了信息隐藏这一原则。接口是模块的外部特征，应当公开；数据结构、算法、实现体等则是模块的内部特征，应当隐藏。

（2）高内聚

- 内聚（Cohesion）是一个模块内部各成分之间相关联程度的度量。模块设计者没有必要确定内聚的精确级别，重要的是尽量争取高内聚，避免低内聚。

（3）低耦合

- 耦合（Coupling）是模块之间依赖程度的度量。内聚和耦合是密切相关的，与其他模块存在强耦合的模块通常意味着弱内聚，而强内聚的模块通常意味着与其他模块之间存在弱耦合。
- 模块设计应当争取“高内聚、低耦合”，而避免“低内聚、高耦合”。

3. 模块设计流程

模块设计的核心工作是“接口设计”和“数据结构与算法设计”。前者是模块的外部特征，应当公开，而后者是模块的内部特征，应当隐藏。

模块设计属于系统设计的详细设计阶段，那么模块设计应当详细到什么程度呢？

- 由于现代的软件开发工具越来越先进，模块的详细设计和编程可以很好地融合在一起，而且效率相当高，有些开发工具甚至具有代码自动生成的功能。所以模块设计究竟要详细到什么地步，应当视问题的复杂性，以及所采用的开发工具而定。
- 一般地，只要确定了每个模块的主要接口、数据结构与算法，能够清楚地指导模块编程即可。总之，不必花太多时间用于设计模块的细节。

4. 超类设计

用面向对象方法开发出的软件具有较强的可重用性，这种重用包括开发项目内部的重用和外部的重用。重用依存于超类设计，没有超类的对象系统好比“把洗衣机当米缸”，不能物尽其用。

由于系统开发工期紧、工作量大，而我们的开发队伍年轻，经验和人力都不足，内部重用率高的超类开发无疑是我们的救星。它可以减少重复劳动，易于统一规格，对复杂问题统一攻关、统一解决，便于统一维护。

对超类的抽象即实例对象的泛化原则：

- 寻找大多数实例对象的共同行为。例如，“打印报表”、“查询静态代码表”、“录入数据库表数据”等。
- 超类的多态性设计要保证使用超类继承关系可以满足各子类的操作要求。例如，继承同一个“数据录入”祖先窗口，可以完成不同结构数据库表的数据录入。
- 利于信息的隐蔽性，不会破坏数据的完整性，利于将复杂问题简单化。
- 有利于推行开发规范，统一界面风格。

5.5.4 数据结构和算法设计

1．设计理念

- 每一种数据结构与算法都有其时间、空间的开销和收益。当面临一个新的设计问题时，设计者要彻底地掌握怎样权衡时空开销和算法有效性的方法。这就需要懂得算法分析的原理，而且还需要了解所使用的物理介质的特性（例如，数据存储在磁盘上与存储在内存中，就有不同的考虑）。
- 与开销和收益有关的是时间—空间的权衡。通常可以用更大的时间开销来换取空间的收益，反之亦然。时间—空间的权衡普遍地存在于软件开发的各个阶段中。
- 程序员应该充分地了解一些常用的数据结构与算法，避免不必要的重复设计工作。
- 数据结构与算法为应用服务。我们必须先了解应用的需求，再寻找或设计与实际应用相匹配的数据结构。

2．一般流程

- 数据结构与算法有全局和局部之分，先设计全局的，后设计局部的（通常在模块设计时进行）。
- 根据问题的特征，先查找已经存在的数据结构与算法，挑选最合适的（并不一定是最先进的）。如果不存在现成的，那么自己设计。
- 设计并且编写代码之后要进行测试，如果不满足性能要求，那么要进一步优化数据结构和算法。

5.5.5　OOP（Object Oriented Programming）设计方法步骤

（1）类的抽象

类的抽象也就是对象建模。简单地说，就是为了实现我们的目的（模块功能）需要有哪些类？一般情况下，我们设计的类是现实具体对象的一个抽象，如“自行车”；但也可能是一个抽象的类（如超类）。出发点可以是该模块的功能（功能划分），或是和用户的交互入口。

（2）类的层次构造

类的层次构造就是这些类的关系是什么？用对象模型图表示出来。即各个类的关系，是继承还是依赖等。

（3）类的定义

类的定义即每个类包含哪些属性（类的成员变量）、哪些操作（类的成员函数）。这是一个事件识别和操作识别的过程。

（4）类的实现

类的实现即具体到每个类的成员函数是如何实现的。对于每个成员函数类似结构化程序设计中的一个模块，要遵守如单入口、单出口；功能单一；规模适中；接口简单等原则。

第6章　编　　码

本章导读：作为一名程序员，首先要精通一种面向对象的编程语言，熟练掌握相应的开发工具；随着网络技术的飞速发展，有很多的网络编程技术如XML、Ajax等都需要我们去了解和学习；作为项目中的一员，了解编码规范和源代码控制工具也是程序员所必须掌握的。在本章中，我们将探讨这些话题。

6.1　编程语言和开发工具

6.1.1　编程语言

伴随几代程序员50多年的编程历史，计算机程序设计语言也已经历了50多年的衍变，目前已经有超过2500种的语言。

面对如此繁多的编程语言，作为一个程序员，到底要学习多少种呢？

正如开篇第1章所述，笔者认为，其实无论哪种语言，作为一个程序员，首先应该熟练掌握并精通一种。当然在目前的环境中，推荐大家首先掌握一种面向对象的语言，其中的典型代表就是Java和C#。

1．Java

Java是Java程序设计语言和Java平台的总称，由Sun公司在1995年推出。Java的出现推动了Internet技术的发展，常用的浏览器现在都支持Java applet。

Java最大的魅力在于跨平台。现在Java平台已经嵌入到几乎所有的操作系统。这

样 Java 程序只需要编译一次，就可以在各种系统中运行。

Java 应用编程接口已经从 1.1x 版发展到 1.2 版。目前常用的 Java 平台基于 Java1.6，最新版本为 Java1.7。可以从 http://www.java.com/下载最新版本。

Java 语言是一种简单的、面向对象的、分布式的、解释型的、健壮安全的、结构中立的、可移植的、性能优异、多线程的动态语言。

Java 分为三个体系，即 J2SE（Java2 Standard Edition），J2EE（Java 2 Platform，Enterprise Edition），J2ME（Java 2 Micro Edition）。

2．Microsoft .NET 和 C#

（1）Microsoft .NET

Microsoft.NET 是微软提出的一种战略，它是一种面向网络、支持各种用户终端的开发平台环境。微软的宏伟目标是让 Microsoft.NET 彻底改变软件的开发方式、发行方式、使用方式等，改变终端用户和商业的计算模式。

如同 MS-DOS 和 Windows 一样，.NET 将大大改变我们的计算领域。MS-DOS 使得个人电脑在商业和家庭中广为接受；Windows 增强了用户的图形界面，使其成为首选的与软件交互方式，最终使得图形界面成为个人电脑的主流。而.NET 则要把 XML Web 服务变成日后的主流计算模式。

.NET 有两个主要的部件：一个是 CLR（Common Language Runtime），另一个是.NET Framework class library。

- CLR 是.NET 系统构架中最核心部分，负责执行代码、管理内存、提供安全性等核心功能，类似于 Java 虚拟机。
- .NET Framework 是在 Microsoft .NET 平台上进行开发的基础，它实现了语言开发、代码编译、组件配置、程序运行、对象交互等各个层面的功能，为 Web 服务及普通应用程序提供了一个托管、安全、高效的执行环境。

.NET 优点包括：

- 强壮，安全的执行环境
- 统一的编程模式
- 统一的编程 API
- 统一的简单开发方式

- .NET 支持多种语言的互操作，即在一种语言下开发的组件，可在另一组件下通过面向对象的继承而得以重用，目前.NET 支持的语言达二十多种。

（2）C#

C#是微软开发的一种面向对象语言，其目标是既拥有 C++的执行效率和运算能力，也具备如 VB 一样的易用性。C#是基于 C++的一种语言，同时包含类似 Java 的很多特征。

C#是专门为.NET 应用而开发出的语言，这从根本上保证了 C#与.NET 框架的完美结合。在.NET 运行库的支持下，.NET 框架的各种优点在 C#中表现得淋漓尽致。通过使用 C#，程序员可以快速地编写各种基于 Microsoft .NET 平台的应用程序。

6.1.2 开发工具

1. Visual Studio.NET

Visual Studio.NET 是微软.NET 平台下最为强大的开发工具。借助它，开发人员能够快速创建高质量、用户体验丰富而又紧密联系的应用程序。无论是软件开发，还是企业应用程序的部署和发布，Visual Studion.NET 都可以提供近乎完美的解决方案。

Visual Studio .NET 是一套完整的开发工具，用于生成 ASP Web 应用程序、XML Web services、桌面应用程序和移动应用程序。Visual Basic .NET、Visual C++ .NET、Visual C# .NET 和 Visual J# .NET 全都使用相同的集成开发环境（IDE），该环境允许它们共享工具并有助于创建混合语言解决方案。另外，这些语言利用了.NET Framework 的功能，此框架提供对简化 ASP Web 应用程序和 XML Web Services 开发的关键技术的访问。

目前最新版本是 Visual Studio.NET 2008。

2. Eclipse 简介

Eclipse 是一个著名的跨平台的、开发源代码的集成开发环境（Integrated Development Environment）IDE。

Eclipse 的设计思想：一切皆为插件。它自身的核心是非常小的，其他所有的功能都以插件的形式附加到该核心上。由于 Eclipse 中的每样东西都是插件，对于给 Eclipse 提供插件，以及给用户提供一致和统一的集成开发环境而言，所有工具开发人员都具有同等的发挥场所。

Eclipse 最初是由 IBM 公司开发的替代商业软件 Visual Age for Java 的下一代 IDE 开发环境，2001 年 11 月贡献给开源社区，现在它由非营利软件供应商联盟 Eclipse 基金会

（Eclipse Foundation）管理。

尽管 Eclipse 是使用 Java 语言开发的，但它的用途并不限于 Java 语言；通过插件，它可以作为诸如 C/C++、COBOL 和 Python 等编程语言的开发工具。

由于其开放源码，任何人都可以免费得到，并可以在此基础上开发各自的插件，因此越来越受人们的关注。近期还有包括 Oracle 在内的许多大公司也纷纷加入了该项目，并宣称 Eclipse 将来能成为可进行任何语言开发的 IDE 集大成者，使用者只需下载各种语言的插件即可。

目前 Eclipse 的市场占有率逐步提高，已经成为 Java 开发平台中的主流。

Eclipse 可以在其官方网站 http://www.eclipse.org 免费下载该软件的打包文件，解压缩后运行目录下的“eclipse.exe”即可使用。

6.2 网络编程必备知识

如今的软件开发中，B/S 结构的应用程序占据了大半壁江山，即大多数的程序都是网络应用程序。因此作为一个网络应用程序的开发者，程序员必须了解以下一些有关网络编程的基本知识。读者可以查阅相关资料做深入了解。

6.2.1 HTML/JavaScript

提到网页设计、网站开发，最基础的知识就是要了解 HTML 相关知识，它包括：HTML（HyperText Mark-up Language）超文本标记语言、Javascript 脚本语言、DHTML、CSS 样式等。相信读者对这部分内容应该是非常熟悉了。市面上有非常多的相关资料可以找到。

6.2.2 XML

XML（eXtensible Markup Language）也是一种标记语言。它与 HTML 不同，HTML 是专门为描述主页的表现形式而设计的，而 XML 侧重于如何结构化地描述信息（即描述数据）。

XML 最大的特点是可扩展的，它只是提供了一个标准，利用这个标准，用户可以根据实际需要定义自己的标记。

XML 具有以下优点：

- 良好的可扩展性
- 内容与形式的分离
- 遵循严格的语法要求
- 便于不同系统之间信息的传输（即通过它，可以实现跨平台传输数据）
- 具有较好的保值性

从某种意义上说，XML 文件可以被看做是一种数据库，但和其他数据库系统不同，XML 是基于纯文本的、自描述性的，它不需要专门的软件来创建，用户甚至可以用“记事本”这样的编辑器来写自己的 XML 文档。这使得 XML 在网络中迅速普及起来。

现如今，各种语言和开发平台中，基本都内置了对 XML 的支持。显然，掌握基本的 XML 语法知识，熟练进行 XML 编程，也是每个网络程序员需要具备的。

6.2.3 WebService

Web 服务提供了一种软件组件生产和流通的技术和标准，它的出现将从根本上改变应用软件生产和传播的方式。把 Internet 从一个简单提供信息传递的网络变成一个可编程的沟通信息平台，真正实现了“软件即服务”这一理念。

利用 Web 服务，任何公司或者个人都可以廉价、迅速地通过 Internet 向全球用户提供服务，建立全球范围内的联系和合作。用户则可以在 Internet 上提供各种各样的服务中重新选择自己需要的服务内容和方式。如果这种 Web 服务的模式得以实现，将从根本上改变现有企业的商业模式和使用 Internet 的方式。

Web 服务作为未来软件存在的一种形式，极大地影响整个软件行业的发展。现在很多的开发工具和开发语言都能够方便地实现 Web 服务。

6.2.4 Ajax

Ajax 全称为“Asynchronous JavaScript and XML”，即异步 JavaScript 和 XML。Ajax 不是一种技术，它实际上是几种技术，每种技术都有其独特之处，合在一起就成了一种功能强大的新技术。

Ajax 包括：

- XHTML 和 CSS

- 使用文档对象模型（Document Object Model）做动态显示和交互
- 使用 XML 和 XSLT 做数据交互和操作
- 使用 XMLHttpRequest 进行异步数据接收
- 使用 JavaScript 将它们绑定在一起

Ajax 的应用支持以上技术的 Web 浏览器作为运行平台。这些浏览器目前包括：Mozilla、Firefox、Internet Explorer、Opera、Konqueror 及 Safari。但是 Opera 不支持 XSL 格式对象，也不支持 XSLT。

Ajax 最大的优势是可以实现 Web 页面的异步局部刷新。使用 Ajax 的异步模式，浏览器就不必等待用户请求操作，也不必更新整个窗口就可以显示新获取的数据。只要来回传送采用 XML 格式的数据，在浏览器里面运行的 JavaScript 代码就可以与服务器进行联系。JavaScript 代码还可以把样式表加到检索到的数据上，然后在现有网页的某个部分加以显示。

使用 Ajax，用户可以创建接近本地桌面应用的直接、高可用、更丰富、更动态的 Web 用户界面。

在 Java 和.NET 编程环境中，都可以采用 Ajax 技术。

6.3 了解极限编程

ExtremeProgramming（极限编程，简称 XP）是由 KentBeck 在 1996 年提出的，是一种软件工程方法学。

XP 是一个轻量级的、灵巧的软件开发方法；同时它也是一个非常严谨和周密的方法。它的基础和价值观是交流、朴素、反馈和勇气；即，任何一个软件项目都可以从四个方面入手进行改善：加强交流；从简单做起；寻求反馈；勇于实事求是。XP 是一种近似螺旋式的开发方法，它将复杂的开发过程分解为一个个相对比较简单的小周期；通过积极的交流、反馈以及其他一系列的方法，开发人员和客户可以非常清楚开发进度、变化、待解决的问题和潜在的困难等，并根据实际情况及时地调整开发过程。另外，在第二版中 XP 新增加了一个价值：尊重，强调项目组内部、程序员和用户互相尊重，并体现在各个活动中。

与传统的在项目起始阶段，定义好所有需求再费尽心思地控制变化的方法相比，

XP 希望有能力在项目周期的任何阶段去适应变化。

“Extreme”（极限）是指，对比传统的项目开发方式，XP 强调把它列出的每个方法和思想做到极限、做到最好；其他 XP 所不提倡的，则一概忽略（如开发前期的整体设计等）。

6.4 编码规范

代码具有良好的可读性，是程序员基本的素质需求。良好的编写习惯，不但有助于代码的移植和纠错，也有助于不同技术人员之间的协作。在一个规范的项目组中，代码的变量命名，代码内注释格式，甚至嵌套中行缩进的长度和函数间的空行数字都有明确规定。

在本书的“规范篇”中，我们给大家提供了一个“C#编码规范”样本，供大家参考，其他语言的编码规范制定比较类似，关键是在于整个项目组能够统一并认真贯彻。

6.5 源代码控制

6.5.1 版本管理的必要性

在软件开发过程中，作为一个开发人员您遇到过下面的情况吗？

- 想记录自己每天对程序的重要更新和改变，却没有办法？
- 突然发现前几天的某一处修改是正确的，而且至关重要，但那处修改却已经被后来的修改覆盖了！
- 某个项目组公用的公共类，同时由几个程序员进行维护，结果导致修改混乱，无法弄清哪个是最新版本！
- ……

如果你遇到了以上情况，这就涉及了软件版本管理问题。

众所周知，现在的软件项目通常都是由一个项目小组来共同完成的，所有人员共同参与项目开发的需求分析、设计、编码和测试阶段，并且有专门的测试部分对已经完成的软件进行测试。同时，在软件开发的整个过程中，涉及方方面面的人员，包括开发人

员、测试人员、客户等，所有人员的交流反馈，都可能导致软件的修改，小到源代码中某个变量的改动，大到系统的重新设计，甚至可能是整个需求的变化。

在这个过程中，软件会出现很多版本，同时也不能保证不出现上述的错误修改的情况，因此急需引进一种管理机制，引进一个工具来对软件的版本进行管理。

软件版本管理，从广义上讲不仅包括源代码的版本管理，还包括对整个项目的管理。软件版本管理主要关注以下一些问题：

- 对项目团队进行整体管理
- 如何以一种有效的机制来协调项目组内的成员
- 如何对小组成员各自承担的子模块进行统一管理
- 如何对小组成员各自所做的修改进行统一汇总
- 如何保留修改的轨迹，以便撤销错误的改动
- 对在研发过程中形成的软件的各个版本如何进行标识、管理及差异识辨等

常见的软件版本管理工具有 Microsoft 公司的 Visual SourceSafe 和开源软件 CVS（Concurrent Versions System)。在后面的工具篇中，我们将通过实例的方式，给大家做详细介绍。

这些版本管理工具的工作思路大致相同，基本为在一台服务器上建立一个源代码库，库里可以存放许多不同项目的源程序。每个用户在使用源代码库之前，首先要把源代码库里的项目文件下载到本地，然后用户可以在本地任意修改，最后用相应命令进行提交，由该工具源代码库统一管理修改。

6.5.2 常用源代码控制工具

1．Microsoft Visual Soucesafe 简介

Microsoft Visual SourceSafe 6.0 是 Microsoft Visual Studio 6.0 这个开发产品家族的一员，但同样适用于 VisualStudio.NET 版本的开发工具，以及其他一些开发工具。在这些开发工具中，VSS 可以被集成到开发环境中，通过单击开发工具的相应菜单，就可以实现对源代码的版本控制，充分体现了 VSS 的易用性和强大功能。

VSS 能够有效帮助开发人员对项目文件（包括源代码）进行管理，可以将所有项目相关的源文件以特定的方式存入内置的数据库。项目组内的成员不能对数据库内的文件进行直接修改，而是由 VSS 来将文件复制到各个成员自己的工作目录下进行修改，然

后将修改过的文件提交（Check in）给 VSS，由它进行综合更新。

同时，VSS 也支持文件共享，当项目组内某个成员向 VSS 添加文件后，该文件就被保存在服务器数据库中，其他成员都可以通过 VSS 获得该文件的一个复制。而且所有成员对所有文件的修改，都会被 VSS 记录，从而使得可以恢复任何时刻、任何位置的修改。

VSS 是软件开发版本管理的利器，它使得开发小组的协调变得简单容易而且很直观。

2．CVS 简介

另外一个常用的版本管理工具是 CVS。CVS 是 Concurrent Versions System 的缩写，Concurrent 有并发的，协作的，一致的等含义。CVS 最初由 Dick Grune 在 1986 年 12 月以 Shell 脚本的形式发布在 comp.sources.unix 的新闻组第 6 卷里；1989 年 4 月，Brian Berliner 设计了 CVS 并编写了代码。之后 Jeff Polk 帮助 Brian 设计了 CVS 模块和销售商分支支持。

在 OpenSource 项目中 CVS 应用得最为广泛。

6.5.3　源代码控制规范

在一个项目组中，就源代码控制操作应该有一个共同的规范，否则会出现源代码管理的混乱，影响整个项目的正常进度。

在本书的“规范篇”中，我们给大家提供了一个“Soucesafe 使用规范”样本，供大家参考。

第 7 章　软件测试

本章导读：在某些开发人员眼里，测试很神秘；在某些开发人员眼里，测试很简单；在某些开发人员眼里，测试就是质量。那么软件测试到底是什么？作为程序员来说，和软件测试人员相比，应该了解和掌握哪些方面的测试知识？在本章中我们将着重介绍这些方面的知识。

7.1 软件测试基本概念

软件测试就是在软件交付用户使用或投入运行前，对软件需求规格说明、设计规格说明和编码的最终复审，是软件质量保证的关键步骤。

什么是测试？在 G.J.Myers 的经典著作《软件测试之艺术》（The Art of Software Testing）中，给出了测试的定义：“程序测试是为了发现错误而执行程序的过程”。除此之外，G.J.Myers 还给出了与测试相关的三个重要观点：

- 测试是为了发现程序中的错误而执行程序的过程。
- 好的测试方案是极可能发现至今为止尚未发现的错误的测试方案。
- 成功的测试是发现了至今为止尚未发现的错误的测试。

然而，上面的说明容易让人产生误解，认为“测试=发现 Bug”。需要补充的是，软件测试并不是以查找程序错误为中心，为了更早地发现问题，需要将测试延伸到需求评审、设计审查活动中去，也就是将“软件质量保证”的部分活动归为测试活动。实际上，

在软件开发的实际操作中，常常将软件测试和质量保证合并起来。可见，软件测试是在有限的时间内提供高质量软件的保证，是一个完整、正规的软件开发过程中非常重要的一个部分。

测试分为单元测试、集成测试、系统测试、用户测试几个阶段。

7.1.1　单元测试

单元测试：开发者编写的一小段代码，用于检验被测代码中的一个很明确的功能是否正确。

一般认为，在结构化程序时代，单元测试所说的单元是指函数，在当今的面向对象时代，单元测试所说的单元是指类。

例如，工厂在组装一台电冰箱之前，会对每个元件都进行测试，这就是单元测试。

单元测试与其他测试不同，单元测试可以看做是编码工作的一部分，应该由程序员完成。可以这么说，程序员有责任编写功能代码，同时也就有责任为自己的代码编写单元测试。提交产品代码时也要同时提交测试代码。测试部门可以做一定程度的审核。

对于程序员来说，如果养成了对自己写的代码进行单元测试的习惯，不但可以写出高质量的代码，而且还能提高编程水平。

7.1.2　集成测试

集成测试（也叫组装测试，联合测试）是指在单元测试的基础上，一个应用系统的各个部件的联合测试，以决定他们能否在一起共同工作并没有冲突。它的最简单的形式是，两个已经测试过的单元组合成一个组件，并且测试它们之间的接口。

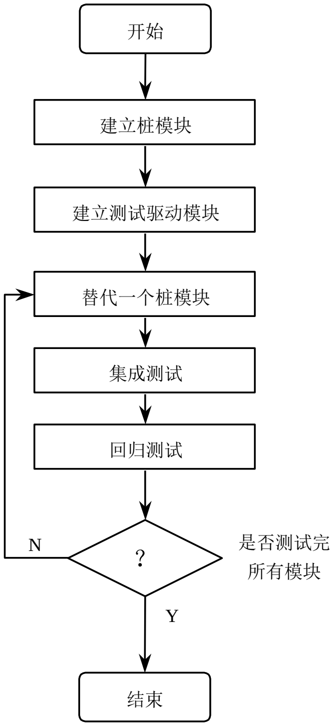
例如，工厂在生产电冰箱时，在完成各个元部件的制作后，要把它们组装起来，同时进行测试看组装是否成功，这就是集成测试。

1．自顶向下集成

自顶向下集成是构造程序结构的一种增量式方式，它从主控模块开始，按照软件的控制层次结构，以深度优先或广度优先的策略，逐步把各个模块集成在一起。深度优先策略首先是把主控制路径上的模块集成在一起，至于选择哪一条路径作为主控制路径，

这多少带有一些随意性，要根据问题的特性确定。

自顶向下集成测试的具体步骤如下图所示。



自顶向下集成示意图

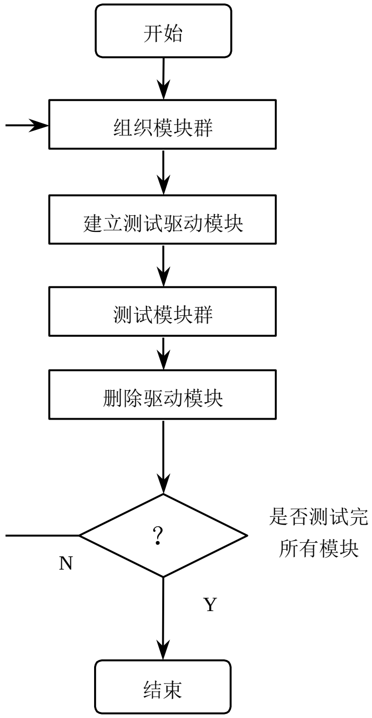
桩模块：能模拟真实模块，给待测模块提供调用接口或数据的测试用软件模块。

建立测试驱动模块：以主控模块作为测试驱动模块，把对主控模块进行单元测试时引入的所有桩模块用实际模块替代。

2．自底向上集成

自底向上集成方式从程序模块结构中最底层的模块开始组装和测试。因为模块是自底向上进行组装的，对于一个给定层次的模块，它的子模块（包括子模块的所有下属模块）事前已经完成组装并经过测试，不再需要编制桩模块。

自底向上集成测试的步骤如下图所示。



自底向上集成示意图

组织模块群：把低层模块组织成实现某个子功能的模块群。

建立测试驱动模块：开发一个测试驱动模块，控制测试数据的输入和测试结果的输出。

7.1.3　系统测试

系统测试是将经过测试的子系统装配成一个完整的系统来测试。目的是验证系统是否满足了需求规格的定义，找出与需求规格不相符合或与之矛盾的地方。

系统测试不仅要测试软件本身，还要测试软件所依赖的其他软、硬件甚至包括某些数据、某些支持软件及其接口等。因此，必须将系统中的软件与各种依赖的资源结合起来，在系统实际运行环境下来进行测试。

7.1.4　性能、压力和负载测试

性能测试就像人和船在无风情况下行驶（正常情况下的性能指标）；主要的性能指

标：服务器的各项指标（CPU、内存占用率等）、后台数据库的各项指标、网络流量、响应时间等。

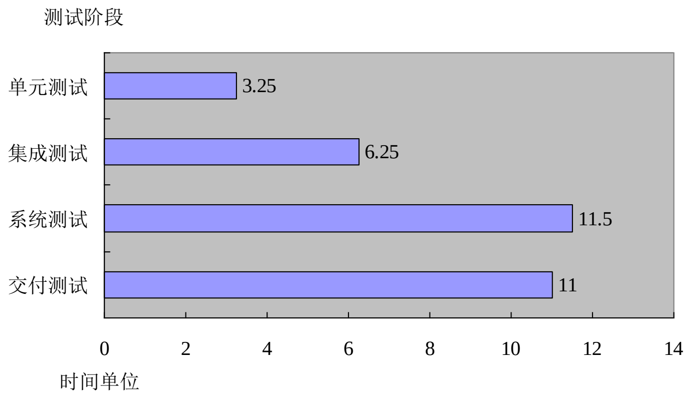
压力测试就像船在微风中行驶（在正常的基础上加大多少百分比压力的性能指标）。

负载测试就像船在强风中行驶（不断加压，直到系统崩溃）。

7.1.5 测试成本

经验表明一个尽责的单元测试方法将会在软件开发的某个阶段发现很多的 Bug，并且修改它们的成本也很低。

下面的图表摘自《实用软件度量》（Capers Jones，McGraw-Hill，1991），它列出了准备测试、执行测试和修改缺陷所花费的时间（以一个功能点为基准），这些数据显示单元测试的成本效率大约是集成测试的两倍、系统测试的三倍（参见条形图）。



针对一个功能点的各类测试所花费的时间统计图

7.1.6 黑盒测试和白盒测试

黑盒测试又称功能测试或数据驱动测试，它完全不管程序内部的结构和处理，把程序看做一个黑盒子，只按程序需求说明书规定的功能和性能测试程序能否正常使用，是否能接收输入数据，产生正确的输出信息。

比如，组装好计算机后，打开电源，启动计算机，测试一下整个系统能否使用，如显示是否正常、声音是否正常等。这可以看做黑盒测试。

白盒测试又称结构测试或逻辑驱动测试，它不管程序的功能，而是根据程序的内部结构设计测试用例，检验程序中每条通路是否都能按预定的要求正确工作。

比如组装计算机时，要分别检测主板、内存条、声卡、显卡、硬盘等各个部件是否正常。这可以看做白盒测试。

7.2 白盒测试用例

测试用例：按一定的顺序执行的，与测试目标相关的测试活动的描述，是确定“怎样”测试。测试用例被看做是有效发现软件缺陷的最小测试执行单元，也被视为软件的测试规格说明书。

7.2.1 测试用例简介

测试用例由测试输入数据以及与之对应的输出结果组成。

测试用例设计的好坏直接决定了测试的效果和结果。所以说在软件测试活动中最关键的步骤就是设计有效的测试用例。

下面的示例代码的开发环境是 VisualStudion.NET，使用的语言是 C#。

```
class myClass
    {
        public int nAge;
        public string strPhase;
        public int Add(int i, int j){return i+j;}
        public void Grow(int years)
        {
            nAge += years;
            if (nAge < 10)
                strPhase = "儿童";
            else if(nAge <20)
                strPhase = "少年";
            else if(nAge <45)
                strPhase = "青年";
            else if(nAge <60)
                strPhase = "中年";
            else
                strPhase = "老年";
        }

    }
    class myClassTester//测试类
    {
```

```
        myClass obj;
        void CaseBegin() { obj = new myClass(); }
        void CaseEnd() { obj = null; }
        public void Test()
        {
            CaseBegin();
            {
//测试用例
                int years = 1;
                obj.nAge = 8;
                obj.Grow(years);
                Assert(obj.nAge == 9);
                Assert(obj.strPhase == "儿童");
            }
            CaseEnd();
        }

        private void Assert(bool p)
        {
            throw new Exception("单元测试报错！");

        }

    }
```

测试用例的核心是输入数据。预期输出是依据输入数据和程序功能来确定的。

测试用例输入数据

	输 入 数 据	说　　明	举　　例
1	正常输入	指正常情况下输入的值	例如：字符串的 Trim 函数，功能是将字符串前后的空格去除，那么正常的输入可以有 4 类：前面有空格；后面有空格；前后均有空格；前后均无空格
2	边界输入	指边界数值	如一个表示年龄的参数，它的有效范围是 0~100，那么边界输入有两个：0 和 100
3	非法输入	非法输入是正常取值范围以外的数据，或使代码不能完成正常功能的输入	如一个进行文件操作的函数，非法输入有这么几类：文件不存在；目录不存在；文件正在被其他程序打开；权限错误

设计测试用例，也分为白盒设计方法和黑盒设计方法。白盒设计方法又分为逻辑覆盖法和基本路径覆盖法，或者分为语句覆盖、判定覆盖、条件覆盖方法，而黑盒设计方法分为等价类划分法、边界值划分法、错误推测法、因果图法等。

这里我们主要介绍白盒测试用例的设计。

7.2.2 白盒测试用例设计

白盒测试用例需要做到：

- 保证一个模块中的所有独立路径至少被使用一次
- 对所有逻辑值均需测试 true 和 false
- 在上下边界及可操作范围内运行所有循环
- 检查内部数据结构以确保其有效性

6 种覆盖方法：语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、组合覆盖、路径覆盖。

1．语句覆盖

语句覆盖的主要特点：要求设计足够多的测试用例，使得程序中每条语句至少被执行一次。语句覆盖是最起码的结构覆盖要求。

优点：可以很直观地从源代码得到测试用例，无须细分每条判定表达式。

缺点：由于这种测试方法仅仅针对程序逻辑中显式存在的语句，但对于隐藏的条件和可能到达的隐式逻辑分支，是无法测试的。

2．判定覆盖

判定覆盖的主要特点：又称为分支覆盖，它要求设计足够多的测试用例，使得程序中每个判定至少有一次为真值，有一次为假值，即程序中的每个分支至少执行一次。每个判断的取真、取假至少执行一次。

优点：判定覆盖比语句覆盖要多几乎一倍的测试路径，当然也就具有比语句覆盖更强的测试能力。同样判定覆盖也具有和语句覆盖一样的简单性，无须细分每个判定就可以得到测试用例。

缺点：往往大部分的判定语句是由多个逻辑条件组合而成的（如，判定语句中包含 AND、OR、CASE），若仅仅判断其整个最终结果而忽略每个条件的取值情况，必然会遗漏部分测试路径。

3．条件覆盖

条件覆盖的主要特点：要求设计足够多的测试用例，使得判定中的每个条件获得各

种可能的结果，即每个条件至少有一次为真值，有一次为假值。

优点：显然条件覆盖比判定覆盖增加了对符合判定情况的测试，增加了测试路径。

缺点：要达到条件覆盖，需要足够多的测试用例，但条件覆盖并不能保证判定覆盖。条件覆盖只能保证每个条件至少有一次为真，而不考虑所有的判定结果。

4．判定/条件覆盖

判定/条件覆盖的主要特点：设计足够多的测试用例，使得判定中每个条件的所有可能结果至少出现一次，每个判定本身所有可能结果也至少出现一次。

优点：判定/条件覆盖满足判定覆盖准则和条件覆盖准则，弥补了二者的不足。

缺点：判定/条件覆盖准则的缺点是未考虑条件的组合情况。

5．组合覆盖

组合覆盖的主要特点：要设计足够多的测试用例，使得每个判定中条件结果的所有可能组合至少出现一次。

优点：多重条件覆盖准则满足判定覆盖、条件覆盖和判定/条件覆盖准则。更改的判定/条件覆盖要求设计足够多的测试用例，使得判定中每个条件的所有可能结果至少出现一次，每个判定本身的所有可能结果也至少出现一次，并且每个条件都显示能单独影响判定结果。

缺点：线性地增加了测试用例的数量。

6．路径覆盖

路径覆盖的主要特点：设计足够的测试用例，覆盖程序中所有可能的路径。

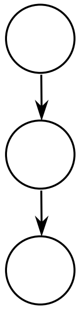
优点：这种测试方法可以对程序进行彻底的测试，比前面 5 种的覆盖面都广。

缺点：由于路径覆盖需要对所有可能的路径进行测试（包括循环、条件组合、分支选择等），那么需要设计大量、复杂的测试用例，使得工作量呈指数级增长。

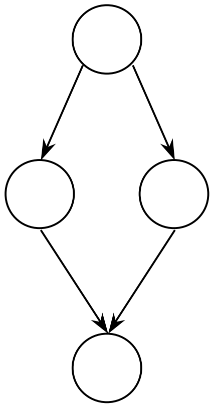
7.2.3　测试用例设计步骤

1．画出程序的控制流图

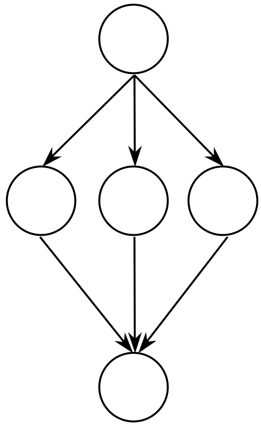
控制流图使用下面的符号描述逻辑控制流，每一种结构化构成元素有一个相应的流图符号。图中的每一个圆称为流图的节点，代表一条或多条语句。流图中的箭头称为边或连接，代表控制流。



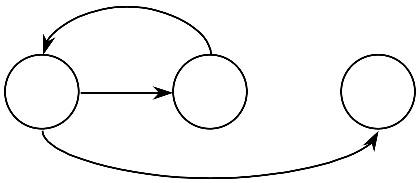
顺序结构



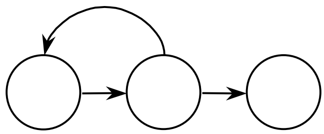
IF 选择结构



CASE 多分支结构



WHILE 循环结构

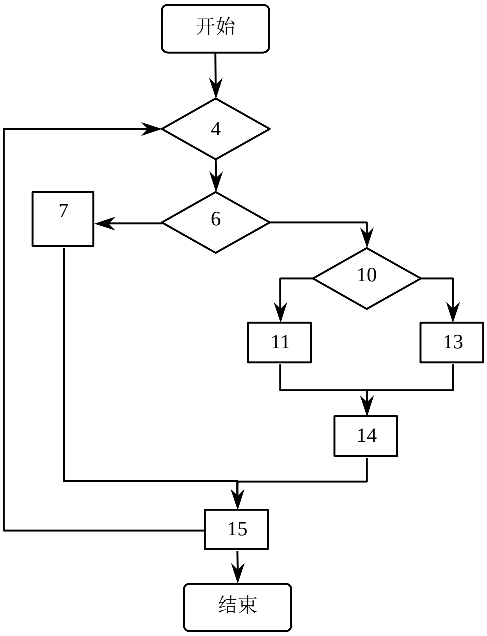


UNTIL 循环结构

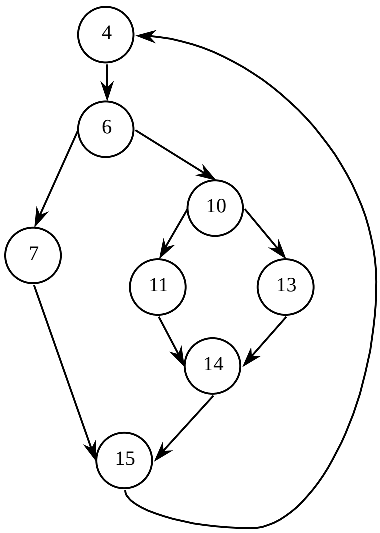
例如，下面是一个用 C#写的函数，为了后面叙述方便，我们在每行前添加了行号。

```
void Sort(int nRecordNum, int nType)
1 {
2     int x = 0;
3     int y = 0;
4     while (nRecordNum-- > 0)
5     {
6        if (nType == 0)
7           x = y + 2;
8        else
9        {
10          if (nType == 1)
11             x = y + 10;
12          else
13             x = y + 20;
14        }
15     }
16  }
```

对应的程序流程图：



对应的控制流图：

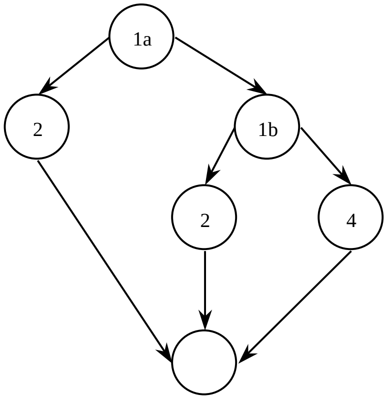


程序设计中遇到复合条件时，生成的流图变得更为复杂。当条件语句中用到一个或多个布尔运算符（逻辑 OR，AND，NAND，NOR）时，就出现了复合条件。下图为语句 IF a OR b 中的每一个 a 和 b 创建了一个独立的节点，包含条件的节点被称为判定节

点，从每一个判定节点发出两条或多条边。例如：

```
if （a or b）
  x
else
  y
```

对应的逻辑图：



2．计算程序圈复杂度

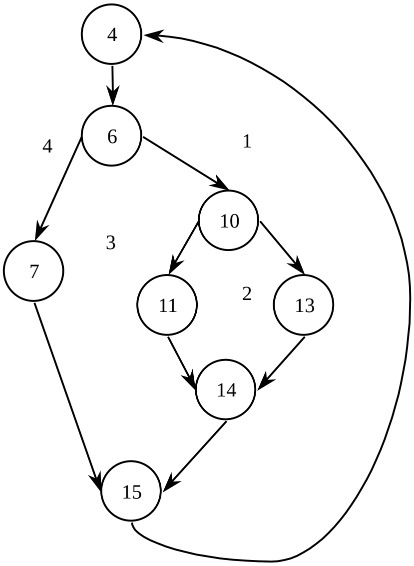
圈复杂度是一种为程序逻辑复杂性提供定量测度的软件度量，将该度量用于计算程序的基本的独立路径数目，为确保所有语句至少执行一次的测试数量的上界。独立路径必须包含一条在定义之前不曾用到的边。

有以下三种方法计算圈复杂度：

- 流图中区域的数量对应于环形的复杂性。
- 给定流图 G 的圈复杂度 V（G），定义为 V（G）=E–N+2，E 是流图中边的数量，N 是流图中节点的数量。
- 给定流图 G 的圈复杂度 V（G），定义为 V（G）=P+1，P 是流图 G 中判定节点的数量。

对应上面图中的圈复杂度，计算如下：

- 流图中有 4 个区域，如下图所示。
- V(G)=10 条边–8 节点+2=4
- V(G)=3 个判定节点+1=4。



3．导出测试用例

根据上面的计算方法，可得出 4 个独立的路径。

- 路径 1：4→15
- 路径 2：4→6→7→15
- 路径 3：4→6→10→11→14→15
- 路径 4：4→6→10→13→14→15

根据上面的独立路径，去设计输入数据，使程序分别执行到上面 4 条路径。

4．准备测试用例

为了确保基本路径集中的每一条路径的执行，根据判断节点给出的条件，选择适当的数据以保证某一条路径可以被测试到，满足上面例子基本路径集的测试用例如下。

- 路径 1：4→15

输入数据：nRecordNum＝0，或者取 nRecordNum<0 的某一个值

预期结果：x＝0

- 路径 2：4→6→7→15

输入数据：nRecordNum＝1，nType＝0

预期结果：x＝2

- 路径 3：4→6→10→11→14→15

输入数据：nRecordNum＝1，nType＝1

预期结果：x＝10

- 路径 4：4→6→10→13→14→15

输入数据：nRecordNum＝1，nType＝2

预期结果：x＝20

5. 工具方法——图形矩阵

导出控制流图和决定基本测试路径的过程均需要机械化，为了开发辅助基本路径测试的软件工具，称为图形矩阵（graph matrix）的数据结构很有用。

利用图形矩阵可以实现自动地确定一个基本路径集。一个图形矩阵是一个方阵，其行/列数控制流图中的节点数，每行和每列依次对应到一个被标识的节点，矩阵元素对应到节点间的连接（即边）。在图中，控制流图的每一个节点都用数字加以标识，每一条边都用字母加以标识。如果在控制流图中第 i 个节点到第 j 个节点有一个名为 x 的边相连接，则在对应的图形矩阵中第 i 行/第 j 列有一个非空的元素 x。

对每个矩阵项加入连接权值（link weight），图矩阵就可以用于在测试中评估程序的控制结构，连接权值为控制流提供了另外的信息。最简单的情况下，连接权值是 1（存在连接）或 0（不存在连接），但是，连接权值可以赋予更有趣的属性：

- 执行连接（边）的概率
- 穿越连接的处理时间
- 穿越连接时所需的内存
- 穿越连接时所需的资源

根据上面的方法对例子画出图形矩阵如下：

	4	6	7	10	11	13	14	15
4		1						1
6			1	1				
7								1
10					1	1		
11							1	
13							1	
14								1
15								

图形矩阵

连接权为“1”表示存在一个连接，在图中如果一行有两个或更多的元素“1”，则这行所代表的节点一定是一个判定节点，通过连接矩阵中有两个以上（包括两个）元素为“1”的个数，就可以得到确定该图圈复杂度的另一种算法。

7.3 软件测试工具

7.3.1 常见软件测试工具列表

常用软件测试工具列表

应用领域		软件名称	所属公司
黑盒测试	功能测试	WinRunner	Mercury Interactive
		QaRun	Compuware
		Rational SQA Robot	IBM
		SilkTest	Borland
		E-Tester	Empirix
	性能测试	Winload	Mercury Interactive
		Qaload	Compuware
		PureLoad	Minq
		WebLode	RadView
		Rational SQA Load	IBM
		LoadRunner	Mercury Interactive
		Web Application Stress Tool	Microsoft

续表

应用领域		软件名称	所属公司
白盒测试	内存资源泄漏检查	Bouncechecker	Numega
		Rational Purify	IBM
	代码覆盖率检查	Truecoverage	Numega
		Rational Purecoverage	IBM
		Logiscope	Telelogic
		Macabe	Macabe
	代码性能检查	Truetime	Numega
		Rational Quantify	IBM
测试管理		Rational Test manager	IBM
		Qadirector	Compuware
		TestDirector	Mercury
		Bugzilla	开源组织
		Trackrecord	Compuware

这里我们只是介绍了常用的部分测试工具，更多的介绍可以参考相关网站和资料。对于程序员来说，主要要掌握一些单元测试工具的使用。

7.3.2　常用单元测试工具介绍

1．xUnit 系列

目前的最流行的单元测试工具是 xUnit 系列框架，常用的根据语言不同分为 JUnit（Java），CppUnit（C++），DUnit（Delphi），NUnit（.NET），PhpUnit（PHP）等。该测试框架的第一个和最杰出的应用就是由 Erich Gamma（《设计模式》的作者）和 Kent Beck（XP（Extreme Programming）的创始人）提供的开放源代码的 JUnit。

- CppUnit

CppUnit 是 C++单元测试工具的鼻祖，免费的开源的单元测试框架。介绍 CppUnit 的资料很多，大家可以通过网络了解这方面的信息。

- JUnit

JUnit 是一个开源的 Java 单元测试框架，用于编写和运行可重复的测试。在 1997 年，由 Erich Gamma 和 Kent Beck 开发完成。

- NUnit

NUnit 是为.NET 框架生成的开放源代码单元测试框架。NUnit 使你可以用你喜欢的

语言编写测试，从而测试应用程序的特定功能。

NUnit 应用程序提供了一个用于编写单元测试的框架，以及一个运行这些测试和查看结果的图形界面。

利用 UNint，我们可以在.NET 编程过程中非常方便地进行单元测试，它的图形化界面和简单而强大的测试框架为我们提供了一个非常舒适而有趣的测试环境，能够让程序员觉得进行单元测试并不枯燥乏味，习惯后甚至还能成为一种乐趣。

- Dunit

Dunit 是 Xunit 家族中的一员，用于 Dephi 的单元测试。Dunit 是一个免费的测试工具，没有代码覆盖率功能，可以从 http://dunit.sourceforge.net/下载到。

2．Jtest

Jtest 是 parasoft 公司推出的一款针对 Java 语言的自动化白盒测试工具，它通过自动实现 Java 的单元测试和代码标准校验，来提高代码的可靠性。

parasoft 同时出品的还有 C++ test，是一款 C/C++白盒测试工具。

3．Visual Unit

Visual Unit 简称 VU，是国产的单元测试工具，它可以自动生成测试代码、快速建立功能测试用例，程序行为一目了然，极高的测试完整性、高效完成白盒覆盖、快速排错、高效调试、详尽的测试报告。可以参考网址：http://www.unitware.cn。前面所述测试要求：完成功能测试，完成语句覆盖、条件覆盖、分支覆盖、路径覆盖，用 VU 可以轻松实现，还有一点值得一提：使用 VU 还能提高编码的效率，总体来说，在完成单元测试的同时，编码调试的时间还能大幅度缩短。

由于 Java 和.NET 是目前流行的两大编程体系，对于单元测试工具，我们重点介绍 JUnit 和 NUnit 的使用。

第 8 章　结项和产品化

本章导读：编码结束了，软件开发就结束了吗？不！远远没有，还有很多工作要做。制作安装程序、进行软件加密、编写说明书和系统帮助文档，都是在结项和产品化阶段要做的工作。

8.1 安装盘制作

8.1.1 使用 InstallShield

提到制作程序安装盘的工具，首先要想到 InstallShield。InstallShield 是 Acresso 软件公司的产品，安装工具领域事实上的标准，已经成为安全安装软件的标准解决方案。

现在被散发的大部分应用软件的安装程序都是由 InstallShield 开发出来的，根据 Macrovision 公司的数据，到 2006 年为止，世界上总计有 7 亿 5 千万台左右的计算机中使用了由 InstallShield 开发的安装程序。

InstallShield 强大灵活而又简单易用。目前最新的版本是 InstallShield 2009，它支持 Microsoft Windows Vista，包括 Internet Explorer 7、Windows Installer 4 和其他令人耳目一新的新技术。

8.1.2 使用 Visualstudio.NET 发布程序

在 VisualStudio.NET 出现之前，我们制作安装盘似乎只有 IinstallShield 一个选择。但现在，对于 Windows 平台下的应用程序，我们可以选择 VisualStudio.NET 来制作安装

盘。特别是用 VS.NET 开发的应用程序，用 VS.NET 来制作安装盘将变得非常简单。

8.1.3 安装盘制作规范

对于一个公司来讲，整个公司出品的软件，其安装盘应该是统一样式的，要遵守统一的“安装盘制作规范”。在本书的第 23 章中，我们提供了一个模板，供大家参考。

8.2 软件加密

8.2.1 软件加密概述

所谓对软件的加密，就是软件开发者通过一些软、硬件技术手段，来达到限制用户使用的技术。

由于不管采取什么样的加密手段，软件最终要在机器上运行，因此对机器而言，它必须是明文。既然机器可以“看见”这些明文，那么黑客通过一些技术，也可以看到这些明文。因此从理论上，任何软件加密技术都可以破解。只是破解的难度不同而已。

技术上的反盗版的任务就是增加黑客的破解难度。让他们花费在破解软件上的成本，比破解这个软件的获利还要高。这样黑客的破解变得毫无意义。

8.2.2 常见的软件加密技术

1．序列号保护

序列号保护是我们最常见到的一种加密技术。

其基本过程：当用户从网络上下载某个共享软件后，一般都有使用时间上的限制，当过了共享软件的试用期后，必须到这个软件的公司去注册后方能继续使用。注册过程一般是用户把自己的个性化信息（如姓名、生日等）告诉给软件公司，软件公司会根据用户的信息计算出一个序列码，在用户得到这个序列码后，按照注册需要的步骤在软件中输入注册信息和注册码，其注册信息的合法性由软件验证通过后，软件就会取消掉本身的各种限制。

这种加密实现起来比较简单，不需要额外的成本，用户购买也非常方便，在互联网上的软件 80%都是以这种方式来保护的。

验证注册码最基本的方法有两种：

- 一种是按用户输入的个性化信息来生成注册码，再同用户输入的注册码比较，公式表示如下：序列号= Function（用户个性化信息）

但这种方法等于在用户软件中再现了软件公司生成注册码的过程，实际上是非常不安全的，不论其换算过程多么复杂，解密者只需把你的换算过程从程序中提取出来就可以编制一个通用的注册程序。

- 另外一种是通过注册码来验证用户名的正确性，公式表示如下：用户名称= F 逆（序列号）（如 ACDSEE）

这其实是软件公司注册码计算过程的反算法，如果正向算法与反向算法不是对称算法的话，对于解密者来说，的确有些困难，但这种算法相当不好设计。

2．加密狗方法

“加密狗”是一种插在计算机并行口上的软、硬件结合的加密产品。一般都有几十或几百字节的非易失性存储空间可供读/写，现在较新的软件狗内部还包含了单片机。

软件开发者可以通过接口函数和软件狗进行数据交换（即对软件狗进行读/写），来检查软件狗是否插在并行口上，或者直接用软件狗附带的工具加密自己 EXE 文件（俗称“包壳”）。当被加密狗保护的软件运行时，程序向插在计算机上的软件狗发出查询命令，软件狗迅速计算查询并给出响应，正确的响应保证软件继续运行。如果没有软件狗，程序将不能运行。

真正有商业价值的软件一般都用软件狗来保护。

3．将软件与机器硬件信息结合

将软件与机器硬件信息结合的方法分为以下几个步骤。

（1）用户安装软件时，软件从用户的机器上取得该机器的一些硬件信息（如硬盘序列号、BOIS 序列号等），然后把这些信息和用户的一些其他信息（如用户名）等进行计算，得到一个序列号。从而在一定程度上将软件和硬件部分绑定。

（2）安装用户需要把这一序列号用 E-mail、电话或邮寄等方法寄给软件提供商或开发商。

（3）软件开发商利用注册机（软件）产生该软件的注册号，并寄给用户。

这种方法技术的优点有：

- 不同的机器注册码不同。用户获得一个密码只能在一台机器上注册使用软件。不同于目前大多软件采用的注册方法，即只要知道注册码，就可以在任何机器上安装注册。

- 不需要任何硬件或软盘，成本低廉。
- 可让软件在不注册前的功能为演示软件，只能运行一段时间或部分功能。注册后就立即变为正式软件。
- 在使用注册号产生软件（注册机）时可采用密码、密钥盘、总次数限制等方法。

8.3 说明书制作

8.3.1 Word 书籍版说明书

书籍版本的说明书类似于图书的编写，关键是要有统一的 Word 样式模板。

8.3.2 电子说明书制作

当前帮助文件主要有两大类，即早期的 HLP 帮助及现在的 CHM 帮助两种。

1. HLP 格式

HLP 格式是早期的作业系统所使用的帮助档案系统。这种格式对读者的作业系统没有太多的要求，Window 95 及以后的版本都可以执行。现在很多执行于 Windows 平台的软件，其帮助档几乎都是 HLP 格式的。当然，随着微软 CHM 帮助档案系统的推出和作业系统的发展，HLP 格式很可能会被 CHM 所代替。HLP 格式的不足之处是美观程度不够好。由于现在很多人对电子读物的美观程度提出了较高的要求，所以这种格式的电子读物目前在网路上不是很有市场。但是，我们可以肯定的一点是，这种格式的电子读物制作简单、获得方便、无须特别的要求就可以阅读。另外，目前很多软件的帮助档还是使用这种格式，所以在短时间内这种格式还不会退出历史舞台。

HLP 文件制作，需要先按一定格式写 RTF 格式文件。常用的制作工具是 Microsoft Help Workshop。

2. CHM 格式

CHM 格式是微软 1998 年推出的基于 HTML 档案特性的帮助档案系统，以替代早先的 WinHelp 帮助系统，在 Windows 98 中把 CHM 类型的档案称做“已组建的 HTML 帮助档案”。被 IE 浏览器支援的 Javascript、VBScript、ActiveX、JavaApplet、Flash、常见图形档（GIF、JPEG、PNG）、音频视频档（MIDI、WAV、AVI）等，CHM 同样支援，并可以通过 URL 与 Internet 联系在一起。这种格式的电子读物的缺点：要求读者的作业

系统必须是 Windows 98 或 Windows NT 及以上版本，另外，还要求读者的作业系统安装有 Microsoft Internet Explorer 3.0 或以上版本。

最常用的制作 CHM 文件的软件是 Microsoft HTML Help Workshop。该软件是微软开发的 CHM 档案的专业制作工具。其最大的特点是它的易用性和强大的编辑功能。由于此软件是官方发布的软件，所以目前许多人都在使用它制作 CHM 档案。

另外，也有很多的机构和编程爱好者开发了 CHM 制作工具，比如 FAR、Web 2 HTML Help、Quick CHM、Easy CHM 等。

第 9 章　项目组建设

本章导读：随着软件编程技术的发展，团队开发变得越来越普遍，因此如何能够提高整个团队的协作能力、提高整体的开发效率，是每个软件企业都关注的问题。在本章中，我们将探讨有关项目建设方面的问题：项目守则、公共代码积累、工作计划总结等。

9.1 项目组守则

一个成熟的项目组，应该有一个共同的项目组守则，以规范大家的编码、数据库设计、页面设计等方面的行为。

在本书的“规范篇”中，我们给大家提供了一个“项目组守则”样本，供大家参考。

9.2 总结常用代码

在一个项目组中，无论使用哪种开发工具，基本上都会用到一些共同的代码，把这些代码编写成项目组公用的基类或组件，会大大方便程序的编写，提高整个项目开发的进度，也是一个优秀项目团队的良好体现。

因此，强烈建议每个项目组把一些公共的基类、常用的代码放在共享空间，作为整个项目组的共同财富。

在本书的配套光盘中，我们提供了一些“ASP.NET 常用代码”供大家参考。

d:\业务相关\01个人相关\图书\计算机\软件开发这点事\光盘\ASP.NET常用代码总结*.*

↑Name	Ext	Size	Date	Attr
[..]		<DIR>	2008-11-05 19:26	----
[03DataGrid]		<DIR>	2008-11-05 19:26	----
01 打开新的窗口并传送参数	txt	220	2007-02-02 00:01	-a--
02为控件添加客户端事件	txt	80	2004-06-12 11:01	-a--
04WebConfig代码	txt	157	2004-06-12 10:19	-a--
05数据集的操作	txt	474	2004-06-12 10:52	-a--
06数据集和字符串的转换	txt	347	2004-07-15 09:08	-a--
07数据库执行Insert_Update_Delete语句	txt	468	2004-06-12 10:22	-a--
08验证	txt	437	2004-06-18 16:45	-a--
09用户登录参考代码	txt	641	2004-06-12 10:34	-a--
10获取错误信息并到指定页面	txt	178	2007-02-02 00:12	-a--
11清空Cookie	txt	68	2007-02-02 00:13	-a--
12回车转换成Tab	txt	615	2007-02-02 00:31	-a--
13对话框	txt	561	2007-02-02 00:23	-a--
14警告窗口	txt	318	2007-02-02 00:28	-a--
15获得焦点	txt	384	2007-02-02 00:29	-a--
16子窗体返回主窗体	txt	343	2007-02-02 00:30	-a--
17判断是否为数字	txt	586	2007-02-02 00:30	-a--
18Server对象	txt	172	2007-02-02 00:40	-a--
19下载文件的方法	txt	529	2007-02-02 00:42	-a--
20加入客户端验证控件	txt	974	2007-02-02 00:43	-a--
21DataBinding绑定表达式	txt	1,312	2007-02-02 00:45	-a--
22如何在删除并重新安装 IIS 之后修复 IIS 映射	txt	560	2007-02-02 00:42	-a--
Base	cs	5,359	2002-05-26 16:45	-a--

有了这些常用代码，可以有效地加快大家的开发进度，缩短新程序员的上手速度。

9.3 共享编程技巧

对于每种编程环境和编程语言，都有一些特殊的编程技巧。这些技巧会极大地帮助大家进行项目开发。同时，项目组所有成员在项目开发的过程中，总结的各种经验、技巧，也可以共享出来，变成整个项目组共同的财富。

9.4 编写工作计划总结

作为一个职业化的程序员，每日编写计划、总结是一个非常良好的习惯，这也是质量体系的要求。在项目组长的统一安排和管理下，每日完成自己的任务，是保证整个项目顺利进行的根本保障。

每日工作计划、总结的格式不拘一格，可以使用 Mircosoft Outlook，也是可以使用 Microsoft Project 来编写，甚至也可以使用记事本，但贵在坚持。

9.5 代码量统计工具

在项目组中，经常要统计每个人的代码量，在本书配套光盘的“代码量统计工具”目录下，我们提供了几个代码量统计工具供大家使用。

使用方法为：根据代码类型选择 bat 文件。例如要统计所有 C#代码，则把 CountCS.bat 和 FF.exe 两个文件复制到所在工程的根目录下，运行 CountCS.bat 即可。

名称	大小	类型	修改日期
CountASP	1 KB	MS-DOS 批处理文件	2002-6-12 16:15
CountCPP	1 KB	MS-DOS 批处理文件	2001-8-20 15:06
CountCS	1 KB	MS-DOS 批处理文件	2008-11-18 14:30
FF	22 KB	应用程序	1995-7-24 20:14
LineCounter	1 KB	MSInfo 文档	1999-6-28 19:21
WC	9 KB	应用程序	1997-1-14 14:04

当然，大家也可以从网络上下载一些免费工具来进行代码量的统计。

第2篇

本篇导读

在本篇中，我们介绍软件开发过程中需要掌握的11个工具。每个工具介绍都本着能够马上动手操作的原则，读者可以按照书中介绍的步骤来逐步操作，完成相应的案例。这些案例基本都可以在本书的配盘中找到。如果想更加深入了解这些工具，可以参考网络相关介绍和其他资料。

本篇涉及的工具如下：

序号	工具类别	工具名称	描　述	提供方式	下载地址
1	建模工具	Rational Rose	一种面向对象的建模工具，利用这个工具，我们可以建立用UML描述的软件系统的模型	试用版	http://www.14.software.ibm.com/webapp/download/byproduct.jsp?pgel=lnav
2		Power Designer	Sybase公司的数据库建模工具	试用版	http://www.sybase.com/downloads
3	单元测试工具	NUnit	.NET框架单元测试框架	免费开源	http://www.nunit.org
4		JUint	Java单元测试框架	免费开源	http://www.junit.org

工具篇

续上表

5	安装盘制作工具	Install Shield	Macrovision 软件公司的产品，安装工具领域事实上的标准	试用版	http://www.acresso.com/downloads/downloads_5850.htm
6		Visual Studio.NET	微软.NET 开发工具	试用版	http://msdn.microsoft.com/zh-cn/vstudio/products/aa700831.aspx
7	帮助文件制作工具	Microsoft Help Workshop	HLP 文件制作工具	免费	http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=34D35502-4DE9-4676-952C-34CC7F64F098
8		Microsoft HTML Help Workshop	CHM 帮助文件制作工具	免费	http://www.microsoft.com/downloads/details.aspx?FamilyID=00535334-c8a6-452f-9aa0-d597d16580cc&DisplayLang=en
9	源代码控制工具	Microsoft Visual SourceSafe	微软发布的源代码管理工具	试用版	包括于 VisualStudio.NET
10		CVS	源代码管理工具	免费开源	http://www.march-hare.com/cvsnt/
11	流程图绘制工具	Visio	微软 Office 组件之一，绘制图表的工具	试用版	http://www.microsoft.com/downloads/details.aspx?FamilyID=D58116EA-41BB-40FB-8BB9-B80C832A54E5&displaylang=zh-cn

第 10 章　Rational Rose 实例入门

任何软件开发的方法都被某种工具最好地支持着。现在市场中有了很多工具——从最简单的绘图工具到成熟的对象建模工具。

Rational Rose 产品家族被设计为软件开发者提供完整的用于开发客户端/服务器、分布式企业和实时系统环境中满足实际业务需求的牢固的、高效率的解决方案的可视化建模工具集合。Rational Rose 产品共享全体通用的标准，使得希望建立业务流程模型的非程序员和建立应用程序逻辑模型的程序员可以相互理解。Rational Rose 工具的评估版可以通过 IBM 软件公司 Web 站点 www.ibm.com 获取。本文使用的是 Rational Rose 2003。

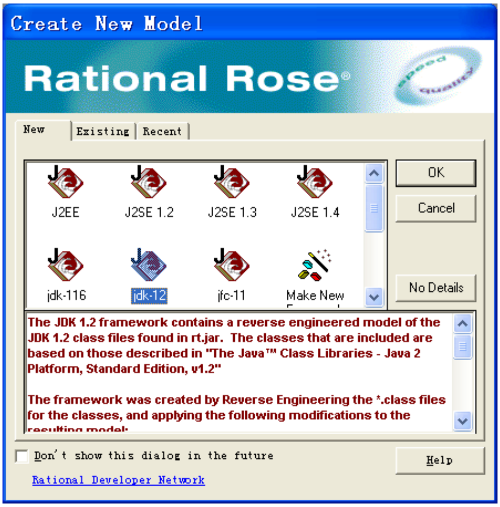
10.1 创建工程

STEP 01 运行 Windows“开始”菜单下“程序”中的“Rational Software\Rational Rose Enterprise Edition”。启动 Rose。第一次启动时，会弹出一个“Create New Model”对话框，如图 10-1 所示。

可以看到有各种版本的模板可供使用。

STEP 02 选择“jdk-12”模板，单击“OK”按钮。

STEP 03 选择“File→Save”，在弹出的“文件保存”对话框中，选择保存文件的路径，输入“TestDemo”。单击“保存”按钮，如图 10-2 所示。



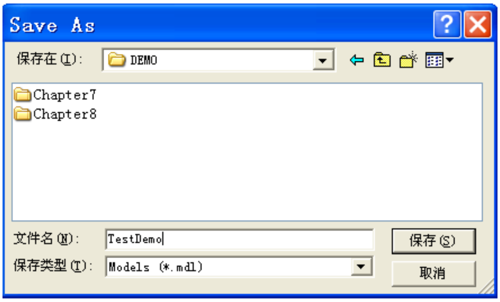


图 10-1 “Create New Model”对话框

图 10-2 文件保存

STEP 04 可以看生成名称为“TestDemo”的工程项目，如图 10-3 所示。

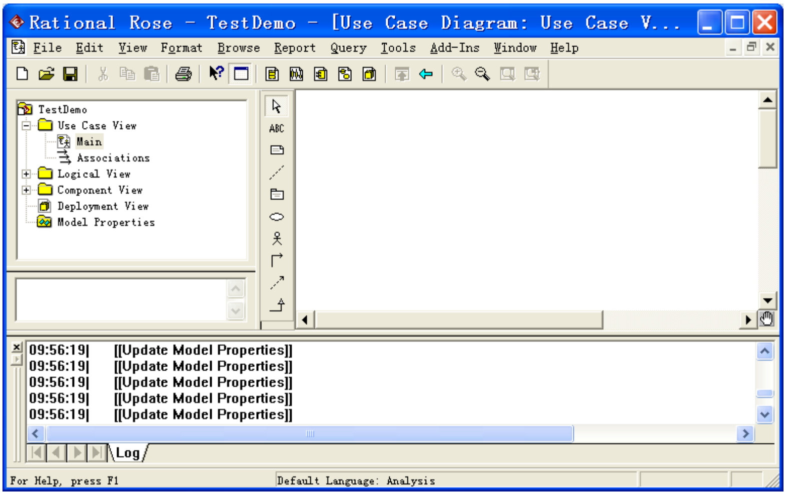


图 10-3 生成项目

可以看到左上方列表中有 4 种视图，它们的含义分别为：

- 使用案例视图（Use Case View）：使用案例是指对系统的一次使用的情况，所有使用案例的集合即构成了系统的全部功能。
- 类和对象视图（Logical View）：类和对象表达了系统的基本结构，类模型确定了

对象的结构细节，是编写程序代码的基础。

- 构件视图（Component View）：在系统的模型中，构件由基本模型中的相应的类组成，通过将类分配到构件中以提供可重用的应用程序结构部件。构件图显示构件之间的依赖关系。
- 配置视图（Deployment View）：配置图显示系统物理设计中各处理模块的分布情况，包括系统硬件的物理拓扑结构，以及执行软件的分布。配置图常用于帮助理解分布式系统。

10.2 建立 Actor

STEP 01 双击“Use Case View”下的“Main”节点；在右边“Use Case Diagram”窗口左侧按钮列中，按下“Actor”按钮。在窗口空白处单击鼠标，生成一个新的 Actor，修改其名称为“SysUser”。可以看到在 Main 节点下同时生成了模型元素“SysUser”，如图 10-4 所示。

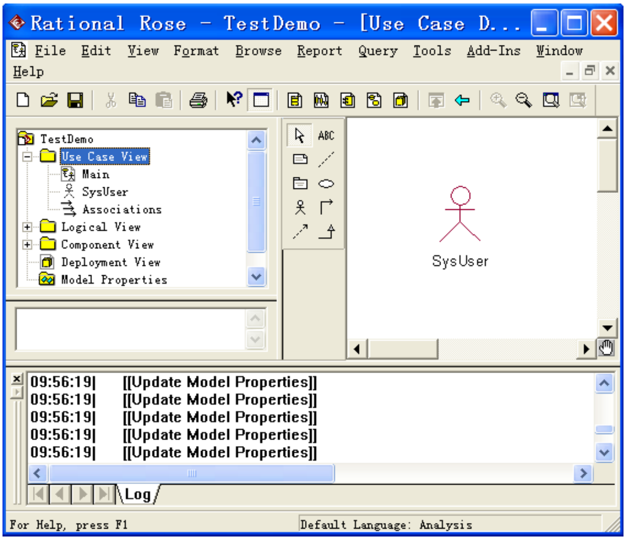


图 10-4　生成建模元素

STEP 02 右键单击“Use Case View”下的“Main”节点，在弹出的菜单中选择“New→Actor”，在“Main”节点下把新生成的模型元素名为“FileSys”。用鼠标左键拖动该元素到右侧窗口，可以看到右侧窗口中添加了一个图形元素，如图 10-5 所示。

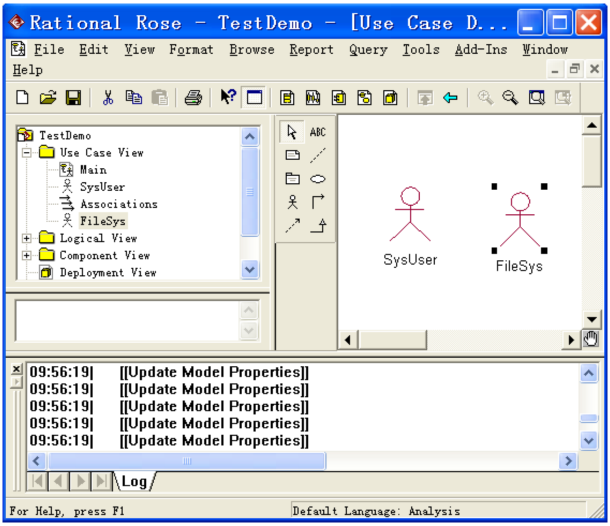


图 10-5　图形元素

10.3 创建 Use Case

STEP 01 创建 Use Case。在右侧窗口中，选中左侧按钮列中的“Use Case”按钮。在空白处单击，创建新的 Use Case，默认名称为“NewUseCase”，修改名称为“Add Task”。同理再添加一个新的 Use Case，命名为“Remove Task”，如图 10-6 所示。

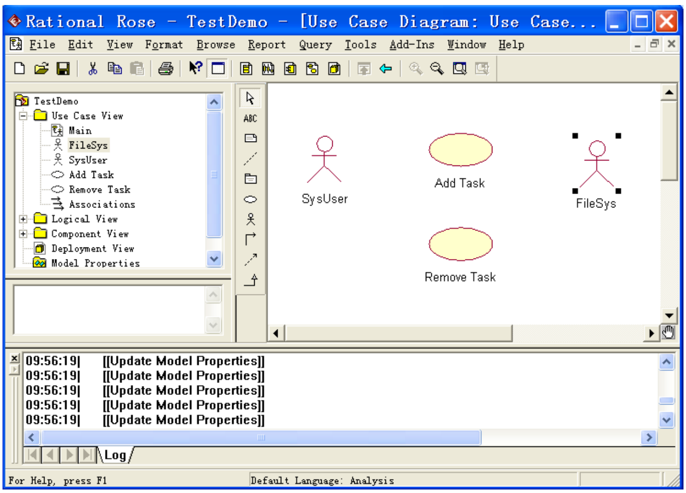


图 10-6　添加新的 Use Case

STEP 02 添加关联。在右侧窗口中，选中左侧按钮列中的“Unidirectional Association”按钮，添加关联如图 10-7 所示。

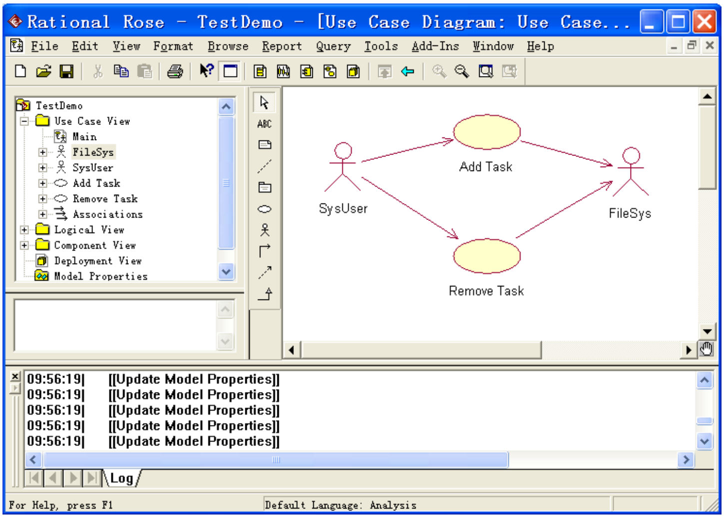


图 10-7 添加关联

10.4 用顺序图描述 Use Case

STEP 01 右键单击左侧窗口中的“Add Task”项，在弹出的菜单中，选择“New→Sequence Diagram”，如图 10-8 所示。

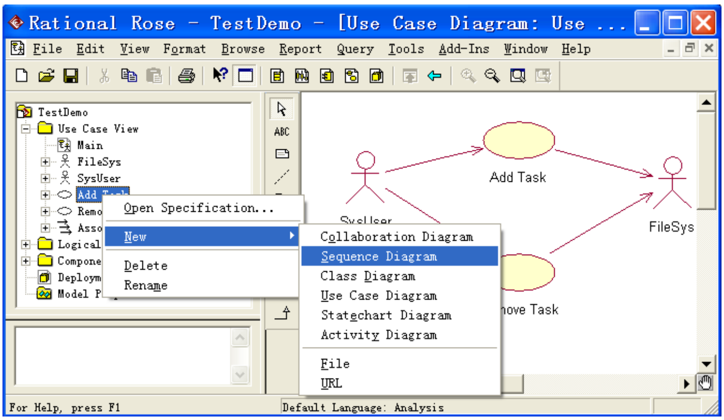


图 10-8 建立顺序图

STEP 02 Rose 会在“Use Case View\Add Task”下新建一项，把它重新命名为“Add a Task”。双击该项，右面会进入“Sequence Diagram”窗口。

STEP 03 从左侧窗口“Use Case View”下，选中“SysUser”项，并拖到右侧窗口，如图 10-9 所示。

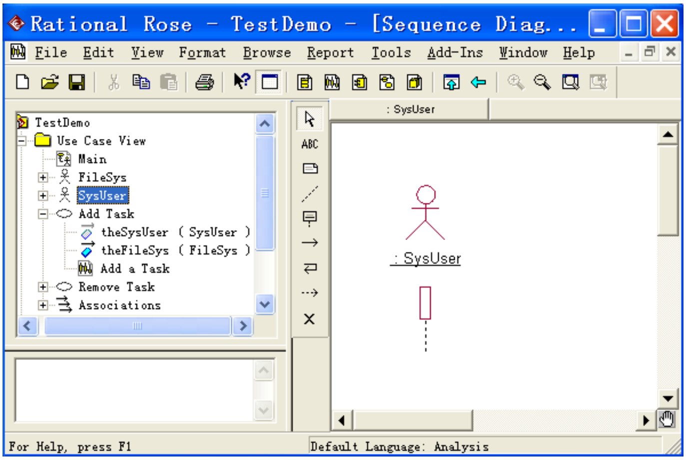


图 10-9　拖图

STEP 04 在右侧窗口中，选中左侧按钮列中的“Object”按钮，添加三个对象，命名为：“FrmTask”、“Tasks”、“Task”。它们表示用户输入窗体、Task 表和 Task，如图 10-10 所示。

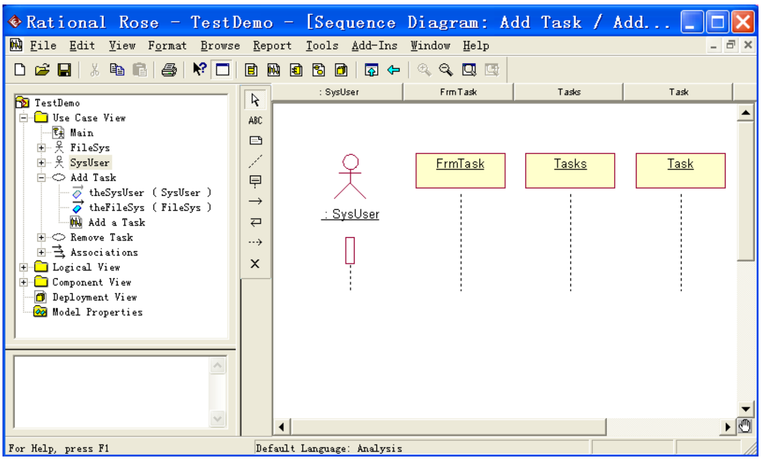


图 10-10　添加对象

STEP 05 添加消息。选中左侧按钮列中的“Object Message”按钮。添加消息如图 10-11 所示。

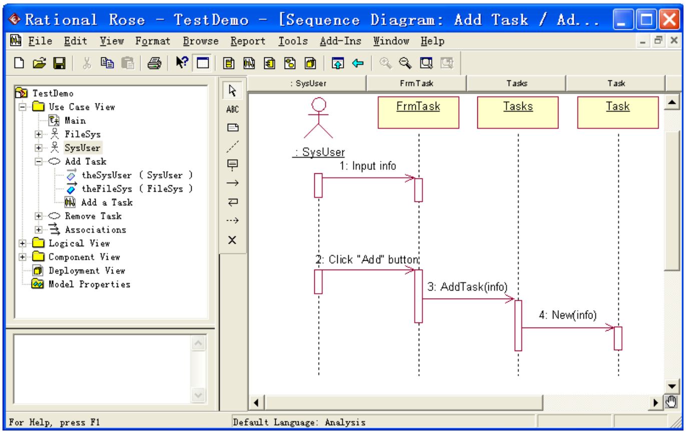


图 10-11　添加消息

STEP 06 选中“Message to Self”工具，为“FrmTask”对象添加一个名为“Add a item into ListBox”自我循环消息。

10.5 建立协作图

STEP 01 选择“Browse→Create Collaboration Diagram”菜单项，如图 10-12 所示。

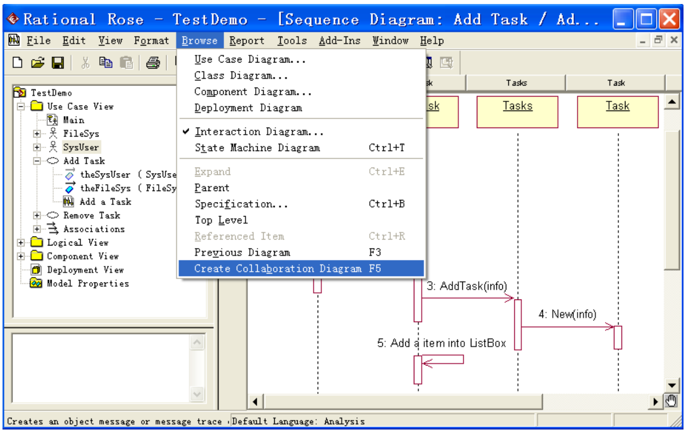


图 10-12　选择菜单

STEP 02 可以看到生成了协作图，如图 10-13 所示。

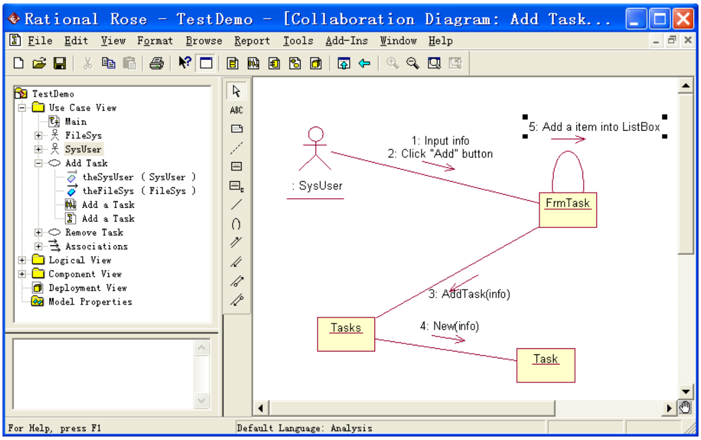


图 10-13 生成协作图

10.6 建立类

STEP 01 双击“Logic View”中的“Main”节点。在工具面板中，选择“Class”按钮，添加三个类，对应上面建立的三个对象，并把它们依次命名为“CFrm”、“CTasks”、“CTask”。

STEP 02 在右侧窗口中，选中左侧按钮列中的“Unidirectional Association”按钮，添加关联，如图 10-14 所示。

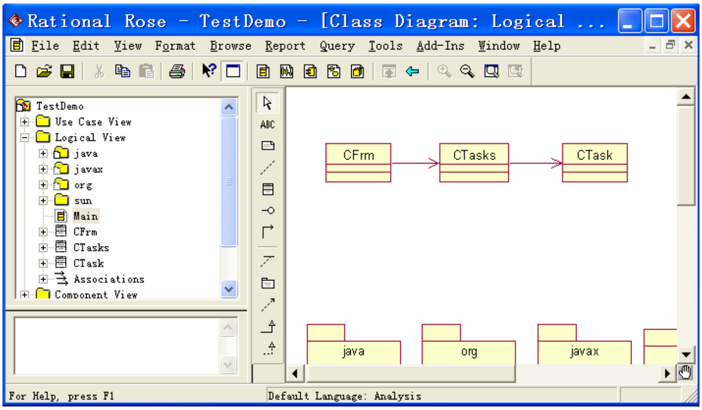


图 10-14 添加关联

STEP 03 切换到刚才创建的属性图。双击“FrmTask”对象，弹出新对话框。在“Class：”下拉列表中，选择“CFrm”，如图 10-15 所示。同理修改其他两个对象。

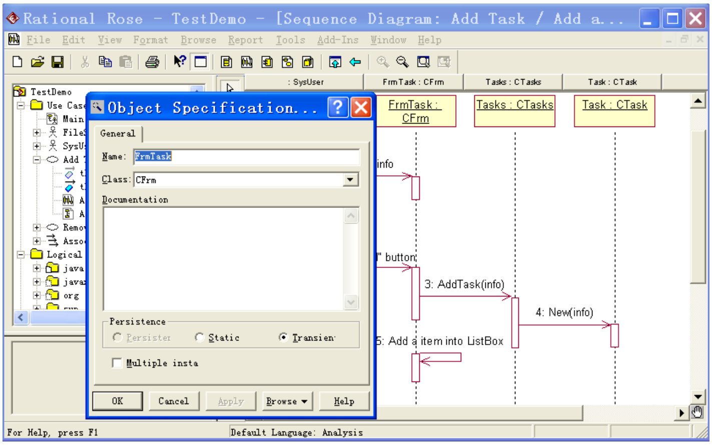


图 10-15 修改对象

接下来，读者可以自己细化类的设计。

10.7 实现模型

STEP 01 双击“Component View”中的“Main”节点。在工具面板中，选择“Component”按钮，添加一个部件。双击该图标，在弹出的对话框上，修改其“Name”和“Stereotype”属性如图 10-16 所示。

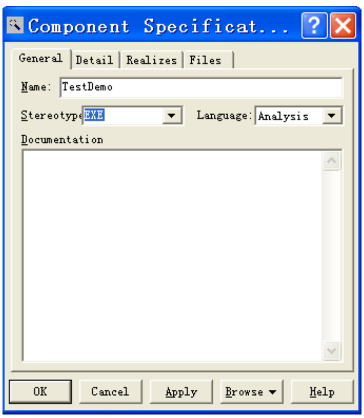


图 10-16 修改属性

STEP 02 添加实现部件。在左侧窗体“Logic View”节点下，选中“CFrm”，并把它拖动到“TestDemo”图标上。可以看到“CFrm”节点变为了“CFrm（TestDemo）”。同理操作“CTasks”和“CTask”两个类，如图 10-17 所示。

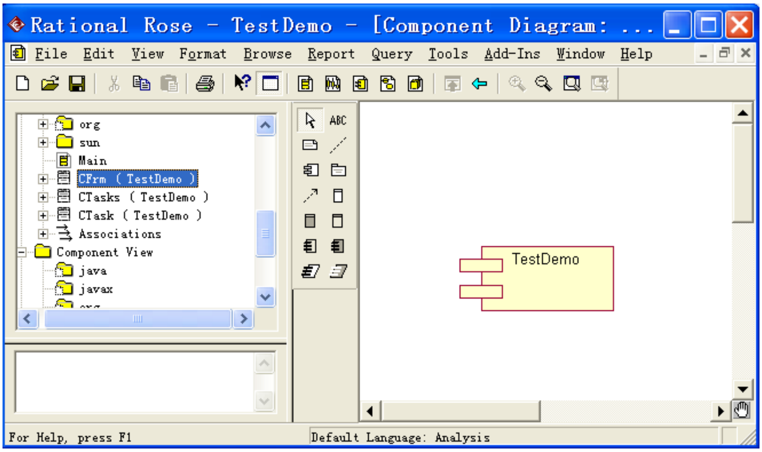
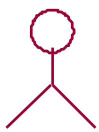


图 10-17　添加实现部件

10.8 UML 符号和图形

在 UML 元模型中定义了很多模型元素，如：Use Case、对象类、接口、组件等，为了模型的可视化，UML 为每个模型元素规定的特定的图形符号来表示。这些符号的实质是对客观世界的刻画和描述。

1．活动者（Actor）

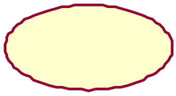


活动者名

活动者是作用于系统的一个角色或者说是一个外部用户。活动者可以是一个人，也可以是使用本系统的外部系统。

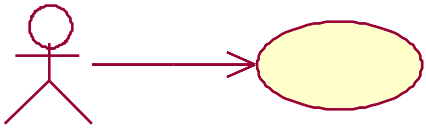
2．用例（Use Case）

用例就是对活动者使用系统的一项功能的交互过程的陈述。



用例名

例如，用户进行登录的用例图可以表示为



用户　　登录

3．对象类（Class）

对象类（类）是具有相同属性和相同操作的对象的集合。



4．接口（Interface）

接口是一种抽象类，它对外提供一组操作，但自己没有属性和方法（操作的实现），它是在没有给出对象实现的情况下对对象行为的描述。接口使用对象类的图形表示方法，接口名前面加构造型“Interface”。



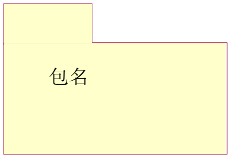
5．组件（Component）

组件体现了系统中逻辑模型元素的物理实现。



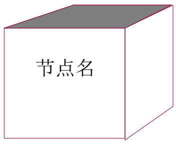
6．包（Package）

包也是一种模型元素，可以把语义相近的模型元素组织在一个包里，增加对模型元素的可维护性。



7．节点（Node）

节点是表示计算机资源运行时的物理对象，一般指有处理能力的硬件设备。节点上可以包含对象和组件的实例。

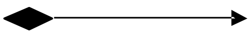


8．关联（Association）

关联就是类或对象之类链接的描述。

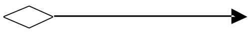
9．组合（Composition）

组合关系用于表示对象之间部分和整体关系，关系很紧密。



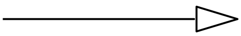
10．聚合（Aggregation）

聚合关系也用于表示对象之间部分和整体关系，但关系比较松散。



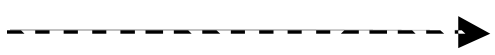
11．泛化（Generalization）

泛化用于表示对象之间一般和特殊的结构关系。



12．依赖（Dependency）

依赖表示两个或多个模型元素之间语义上的关系。

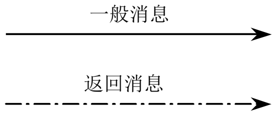


13．实现（Realization）

实现是指一个模型元素（例如，类）是另一个模型元素（例如，接口）的实现。



14．消息（Message）



15．状态（State）

状态描述了对象在生命周期中的一个时间段。



16．注释（Comment）

注释没有特定的语义，它用于对其他模型元素的补充说明。



第 11 章　Power Designer 实例入门

在数据库建模的过程中，PowerDesigner 是最常用的工具之一。运用 PowerDesigner 进行数据库设计，不但能让人直观地理解模型，而且可以充分运用数据库的技术，优化数据库的设计。

在本章中，我们将用一个实际的案例来演示如何利用 PowerDesigner 进行数据库设计。我们使用的版本是 PowerDesigner 12.5。大家可以从 Sybase 公司网站上下载 15 天的试用版。

11.1 创建项目工程

STEP 01 新建工程。在 Windows“开始”菜单“程序→Sybase→Power Designer12”中单击程序项“PowerDesigner”，启动主程序。

STEP 02 选择“File→New”，弹出如图 11-1 所示对话框，选择 CDM 模型（即概念数据模型），在“Modelname”文本框后输入“PDTest”，单击“确定”按钮建立模型。

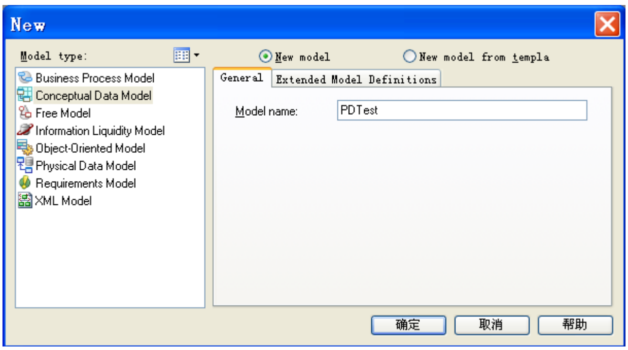


图 11-1 “新建”对话框

STEP 03 选择“File→Save”，弹出另存为对话框，选择要保存文件的路径，如图11-2 所示。

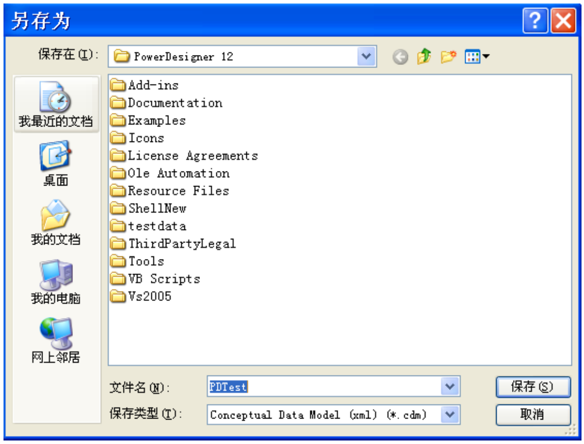


图 11-2 “另存为”对话框

11.2 建立概念数据模型

11.2.1 创建实体

STEP 01 在右侧“图表窗口”中，单击工具箱上的“Entity”工具，再单击窗口的空白处，在单击的位置就出现一个实体符号。单击“Pointer”工具或单击鼠标右键，可以释放 Entity 工具，如图 11-3 所示。

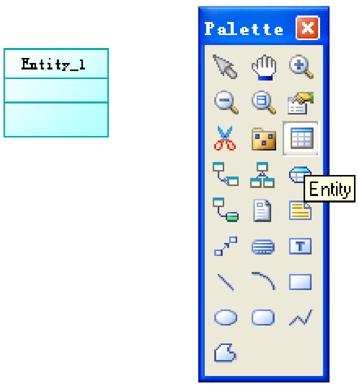


图 11-3 释放 Entity 工具

STEP 02 双击刚创建的实体符号，弹出“实体属性”对话框，选择“General”属性页，在“Name”文本框中输入“Ticket”、“Comment”中输入“飞机票实体”，如图 11-4 所示。

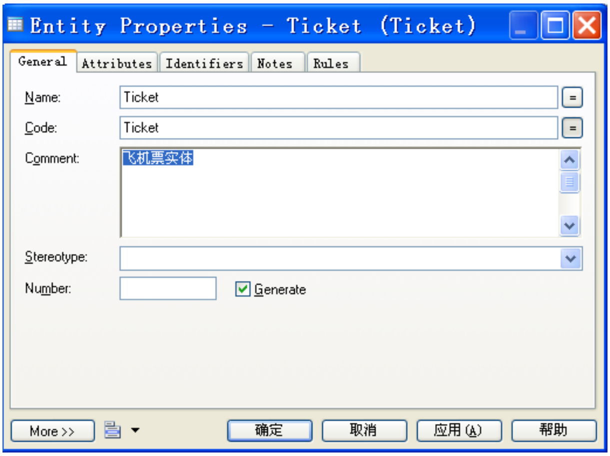


图 11-4 “实体属性”对话框

11.2.2 添加属性

STEP 01 在上述对话框中，选择“Attributes”属性页。单击最左边第一个按钮“Insert a Row”，添加新的属性。修改 Name 为 TicketID，DataType 为 Integer，并把 P、D、M 三个复选框都打钩（P 列表示该属性是否为主标识符，主标识符类似于数据库中的主键；D 列表示该属性是否在图形窗口中显示；M 列表示该属性是否为强制的，打钩则表示不能为空）。

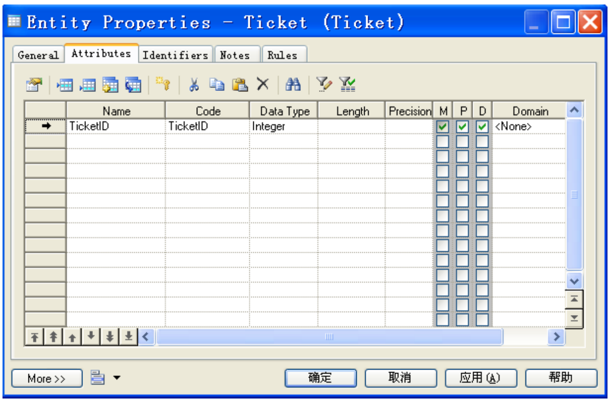


图 11-5 属性页面

STEP 02 同理为“Ticket”实体添加如下属性，如图 11-6 所示。

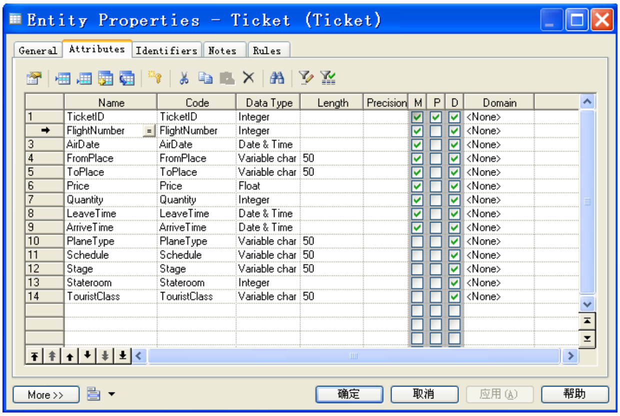
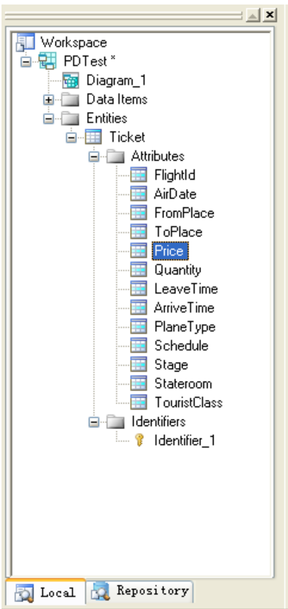


图 11-6 添加属性

11.2.3 定义属性的标准检查约束

STEP 01 在左侧“对象浏览器窗口”中，选择“Price”节点，如图 11-7 所示。

STEP 02 双击该节点，弹出“Price 属性设置”对话框，如图 11-8 所示。



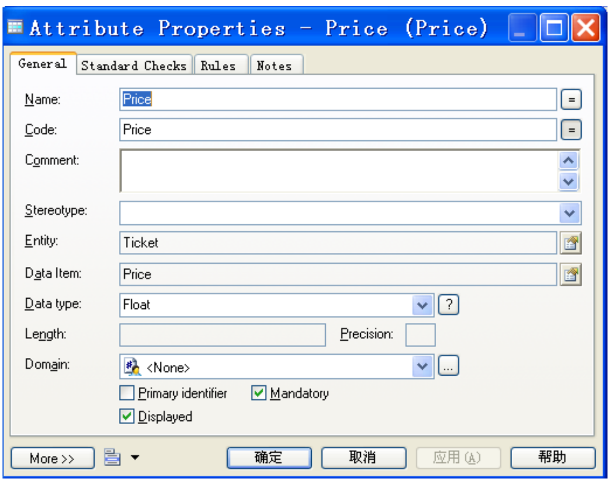


图 11-7 对象浏览器

图 11-8 “Price 属性设置”对话框

STEP 03 在以上对话框中，选择“Stand Checks”属性页。把最小值和默认值都设置为 0，如图 11-9 所示。

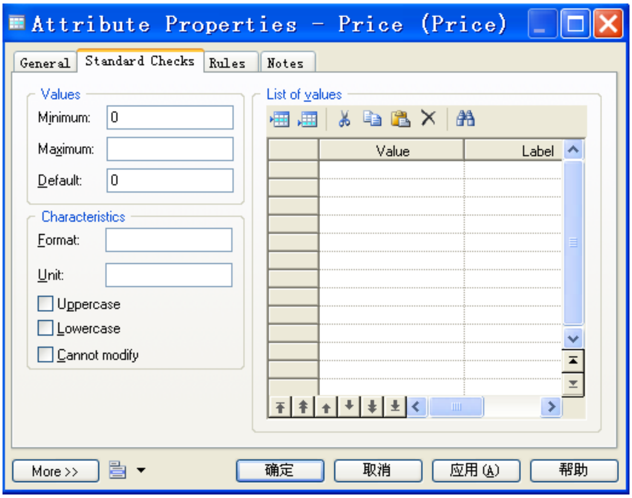


图 11-9 “参数设置”对话框

该窗口中每项的参数的含义如下：

参　　数	说　　明
Minimum	属性最小值
Maximum	属性最大值
Default	系统提供的默认值
Unit	单位，如公里、吨、元
Format	属性的数据显示格式
Lowercase	属性的赋值全部变为小写字母
Uppercase	属性的赋值全部变为大写字母
Cannot modify	该属性一旦赋值不能再修改
List of Values	属性赋值列表，除列表中的值，不能有其他的值
Label	属性列表值的标签

11.2.4　定义实体的主、次标识符

STEP 01 在右侧图表窗口中，双击“Ticket”实体，弹出实体的“属性”对话框。选择“Identifiers”属性页，进行实体标识符的定义。右键单击“Identifier_1”，弹出如图 11-10 所示的菜单。

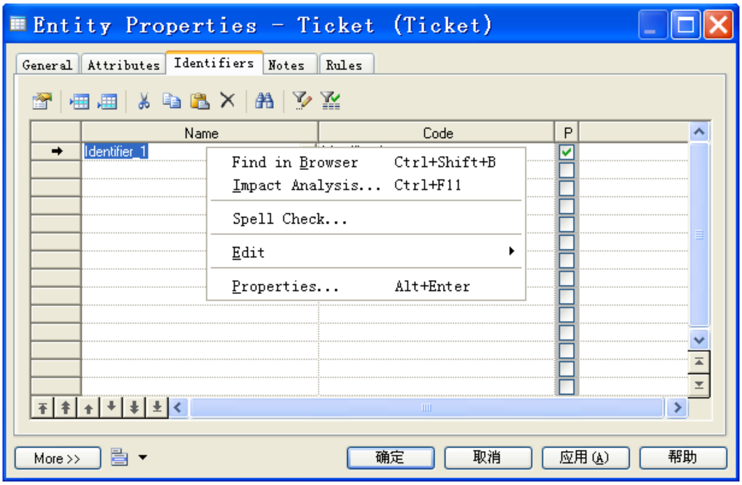


图 11-10 “标识”对话框

STEP 02 在弹出的菜单中选择“Properties…”，弹出“标识属性”对话框，修改标识的名字为“MainIdentifier”，如图 11-11 所示。

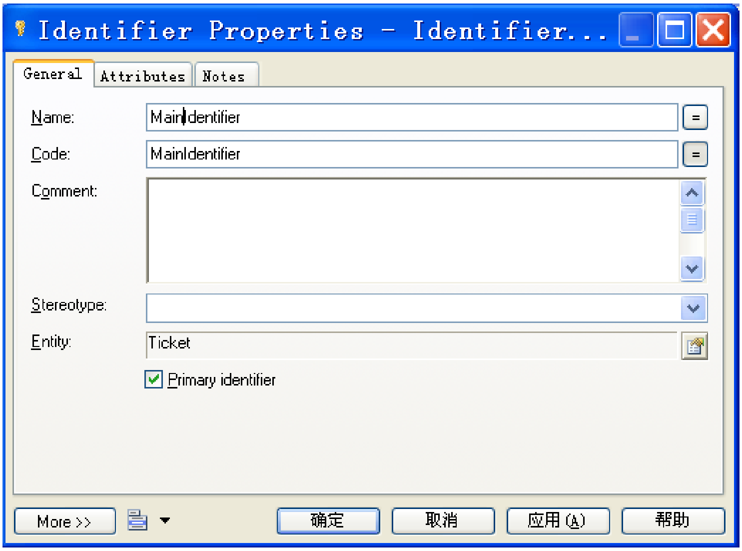


图 11-11 “标识属性”对话框

STEP 03 在以上对话框中，选择“Attributes”属性页。在这一页，可以选择某个属性作为标识符。由于我们在创建属性时设置了 TicketID 为主标识符，所以可以看到 TicketID 显示在表格中，如图 11-12 所示。

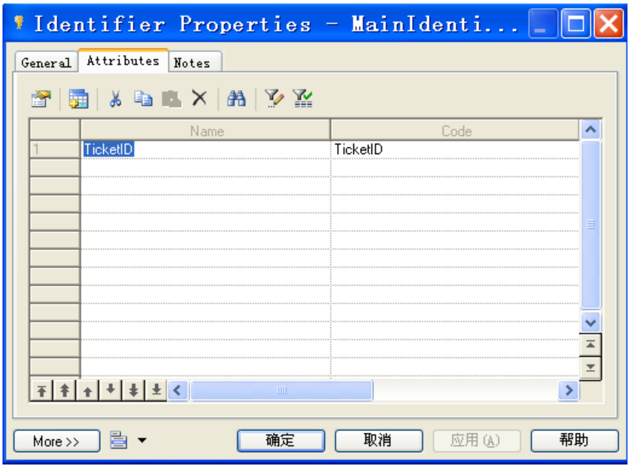


图 11-12　选择“Attributes”属性页

11.2.5　创建其他实体

STEP 01 在右侧“图表窗口”中，单击工具箱上的“Entity”工具，创建顾客实体“Customer”。并为其添加新属性，如图 11-13 所示。

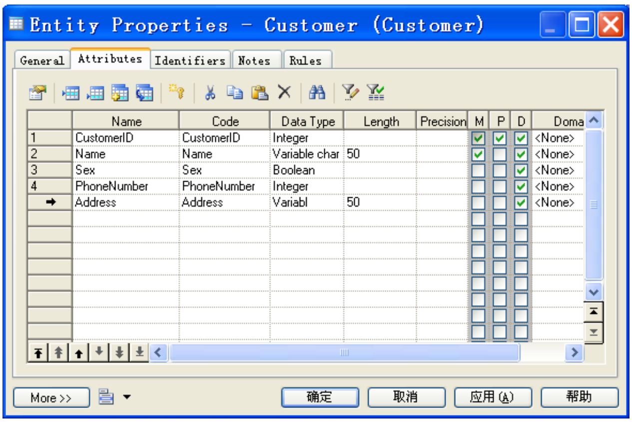


图 11-13　“属性”对话框

STEP 02 同理添加代理商实体“Agent”，并添加如图 11-14 所示的属性。

STEP 03 同理添加销售单实体“SaleList”，并添加相应属性。单击“Reuse Data Item”，可以把以前其他实体中的属性添加进来，如图 11-15 所示。

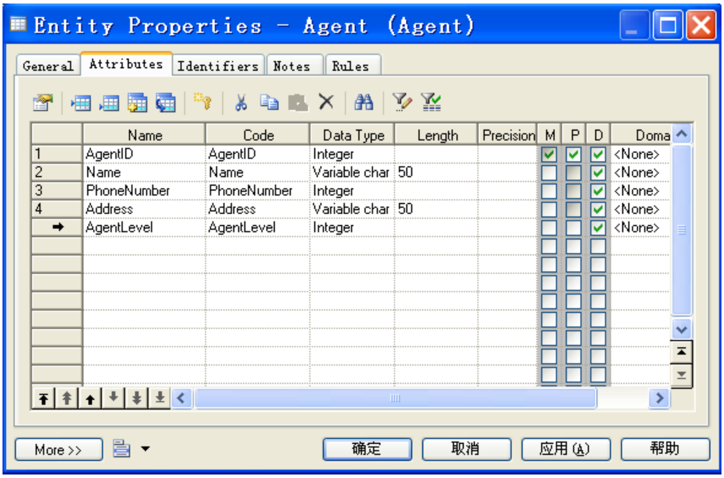


图 11-14 “Agent”实体属性

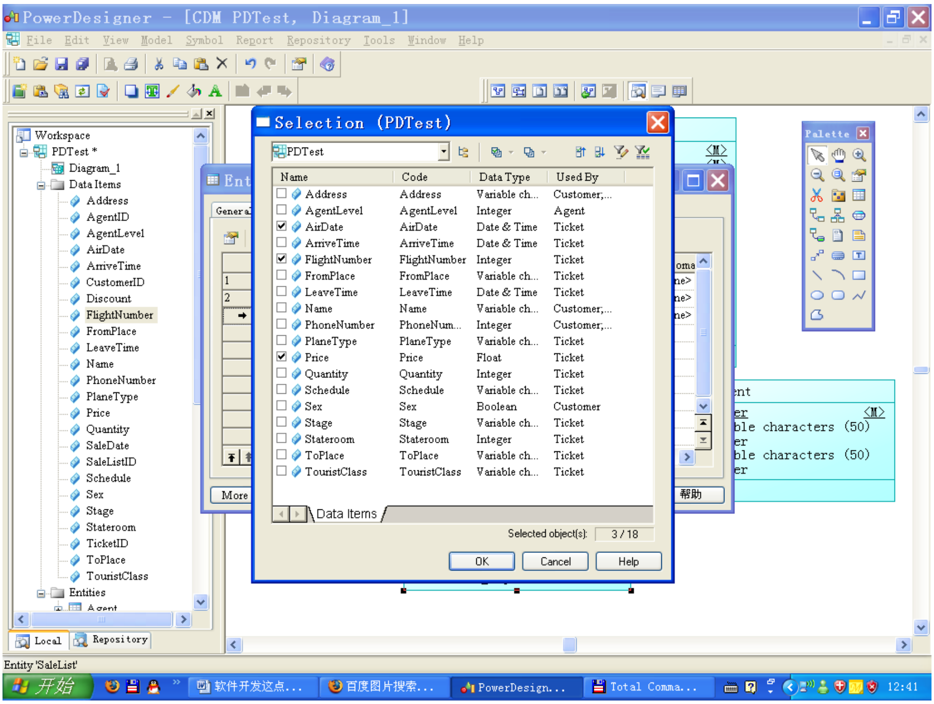


图 11-15 添加实体及属性

添加后的属性如图 11-16 所示。

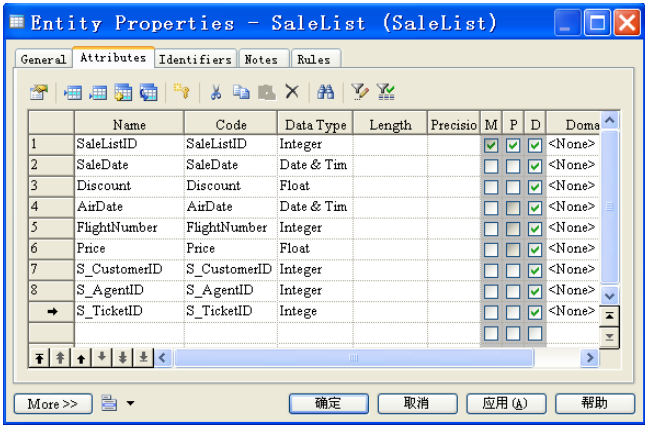


图 11-16 “SaleList”实体属性

建好后，在右侧图表窗口可以看到如图 11-17 所示的样式。

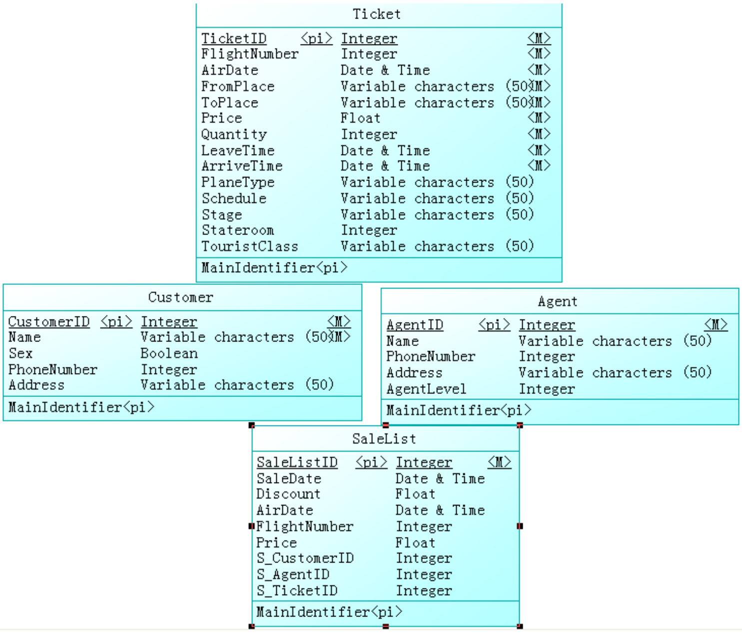


图 11-17 创建的实体样式

11.2.6 建立联系

STEP 01 在图表窗口中，选择工具箱中的“RelationShip”工具，单击“Ticket”实体，在按下鼠标左键的同时把光标拖至“SaleList”实体上并释放鼠标左键，这样就在

两个实体间创建了名字为“Relationship_1”的联系。

STEP 02 双击“Relationship_1”联系线。弹出“联系属性设置”对话框，可以在“Name”文本框中修改该联系名，如图 11-18 所示。

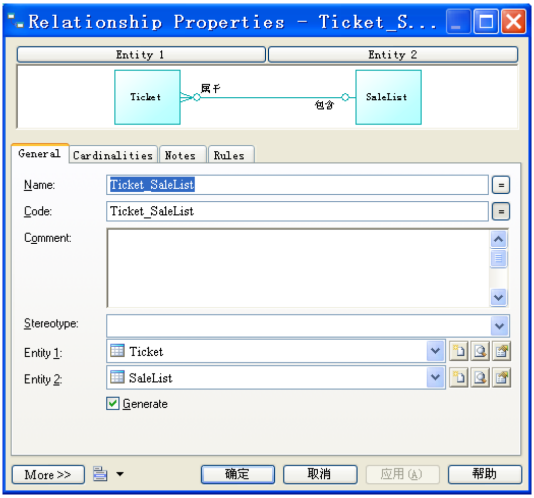


图 11-18 “联系属性”对话框

STEP 03 在以上对话框中，选择“Cardinalities”属性页。在联系类型中，选择“Many-One”（即多对一的关系。因为一张订单中可以包括多张机票），并在两个“Role name”文本框中分别输入“属于”和“包含”，如图 11-19 所示。

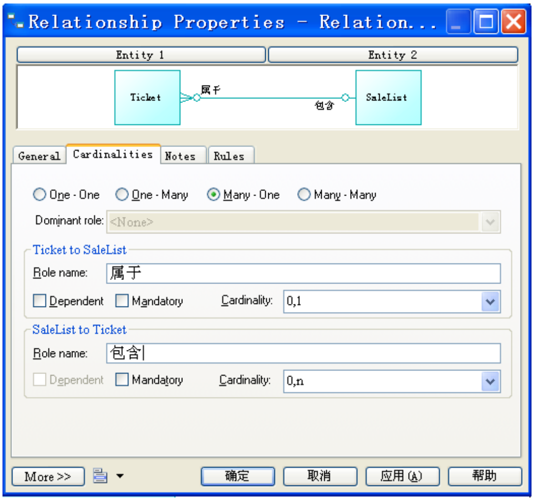


图 11-19 修改联系设置

STEP 04 为“Agent”和“SaleList”实体之间添加联系，命名为“Agent_SaleList”，设置如图 11-20 所示。

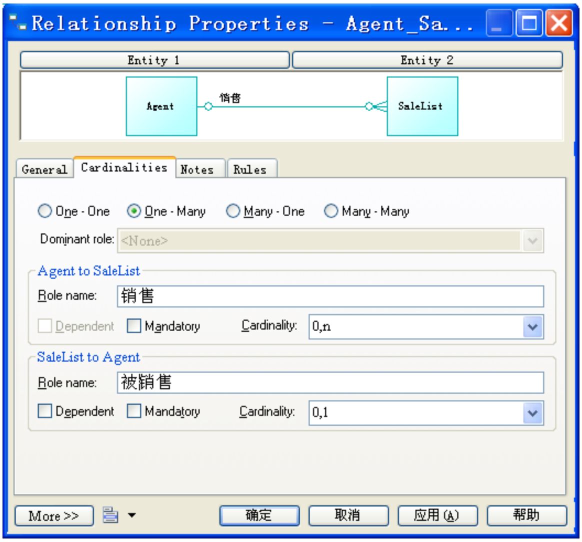


图 11-20 “Agent_SaleList”联系属性设置

STEP 05 为“Customer”和“SaleList”实体之间添加联系，命名为“Customer_SaleList”，设置如图 11-21 所示。

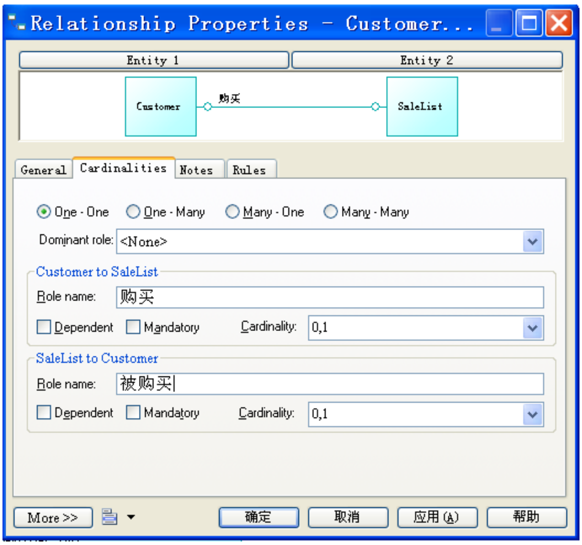


图 11-21 “Customer_ SaleList”联系属性设置

11.3 生成物理数据模型

STEP 01 在系统菜单中选择“Tools→Generate Physical Data Model”，弹出“PDM Gerneration Options”对话框。在 DBMS 数据库选择下拉列表中，选择“Microsoft SQL Server2005”，如图 11-22 所示。

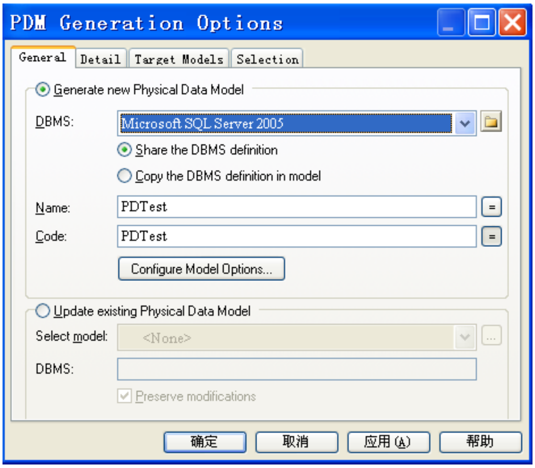


图 11-22 “PDM Gerneration Options”对话框

STEP 02 在以上对话框中，选择“Detail”属性页，如图 11-23 所示。

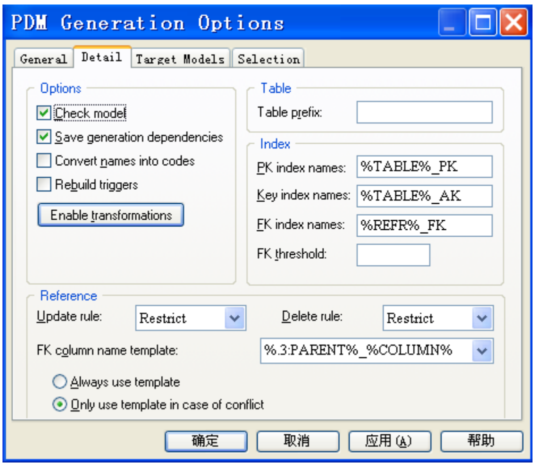


图 11-23 “Detail”属性页

如果选择了“Check Model”，模型将会在生成之前被检查。“Save Generation Dependencies”选项决定 PowerDesigner 是否为每个模型的对象保存对象识别标签，这个选项主要用于合并由相同 CDM 生成的两个 PDM，如图 11-24 所示。

STEP 03 在以上对话框中，选择“Selection”属性页。选择要添加的实体。

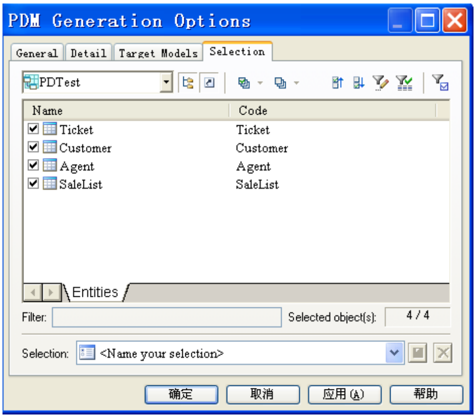


图 11-24 “Selection”属性页

STEP 04 单击“确定”按钮，PowerDesigner 开始生成物理数据库模型。完成后，可以看到在左侧对象浏览器窗口中，新添加的物理数据模型。打开该模型，如图 11-25 所示。

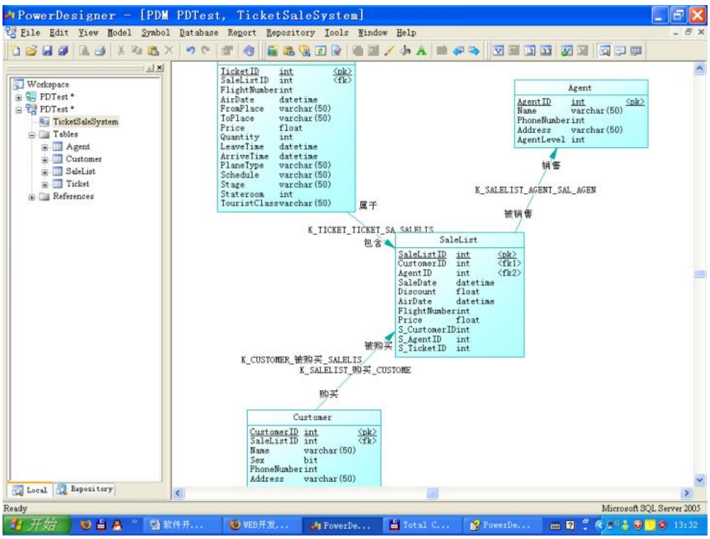


图 11-25 生成的物理数据库模型示意

11.4 生成数据库建库脚本

生成数据库脚本，又称为正向工程。

STEP 01 选择菜单“Database→Generate Database”，弹出数据库生成对话框。在“File name”文本框中输入“TicketSaleSystemSql”，如图 11-26 所示。

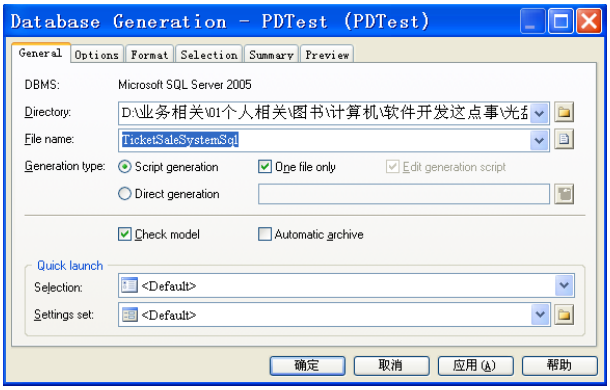


图 11-26 “DataBase Generation”对话框设置

STEP 02 单击“确定”按钮，PowerDesigner 开始生成 SQL 脚本。完成后弹出“Generated Files”对话框，如图 11-27 所示。

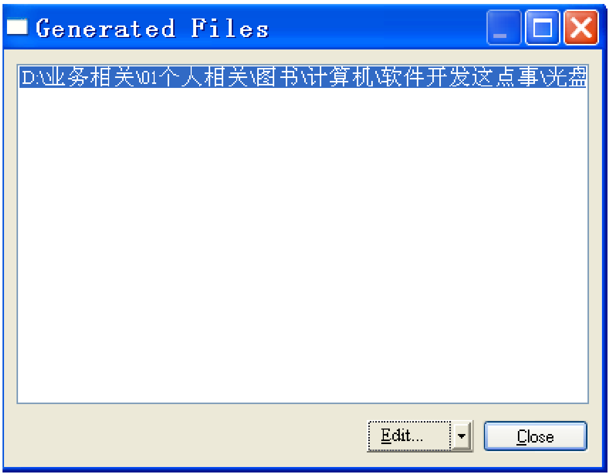
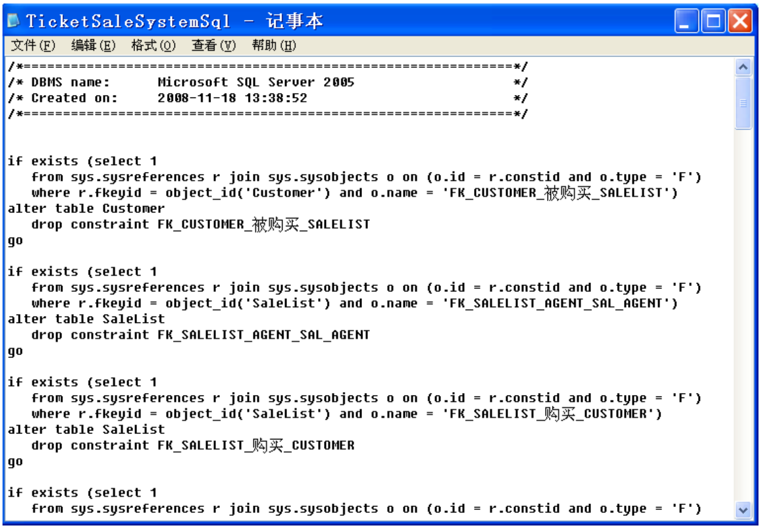


图 11-27 “Generated Files”对话框

STEP 03 单击以上对话框的“Edit…”按钮可以编辑该脚本，如图 11-28 所示。




```
/*==============================================================*/
/* DBMS name:      Microsoft SQL Server 2005                    */
/* Created on:     2008-11-18 13:38:52                          */
/*==============================================================*/


if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
   where r.fkeyid = object_id('Customer') and o.name = 'FK_CUSTOMER_被购买_SALELIST')
alter table Customer
   drop constraint FK_CUSTOMER_被购买_SALELIST
go

if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
   where r.fkeyid = object_id('SaleList') and o.name = 'FK_SALELIST_AGENT_SAL_AGENT')
alter table SaleList
   drop constraint FK_SALELIST_AGENT_SAL_AGENT
go

if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
   where r.fkeyid = object_id('SaleList') and o.name = 'FK_SALELIST_购买_CUSTOMER')
alter table SaleList
   drop constraint FK_SALELIST_购买_CUSTOMER
go

if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
```

图 11-28 生成的 SQL 文本

11.5 生成关系数据库

打开 Microsoft SQL Server 2005，运行刚才生成的 SQL 脚本，数据库创建成功！

本案例生成的 Demo 文件存放在本书配盘“Demo\PowerDesigner”目录下。

11.6 Power Designer 的 4 种模型文件

1．概念数据模型（CDM）

CDM 表现数据库的全部逻辑的结构，它不考虑物理实现细节，只考虑实体之间的关系。一个概念模型经常包括在物理数据库中仍然不实现的数据对象。概念数据模型是最终用户对数据存储的看法，反映了用户的综合性信息需求。

CDM 是适合于系统分析阶段的工具。

2．物理数据模型（PDM）

PDM 用来叙述数据库的物理实现。主要目的是把 CDM 中建立的现实世界模型生

成特定的 DBMS 脚本，产生数据库中保存信息的储存结构，保证数据在数据库中的完整性和一致性。

3．面向对象模型（OOM）

OOM 包含了一系列包、类、接口和他们的关系。这些对象形成了一个软件系统的逻辑的设计视图的类结构。一个 OOM 本质上是软件系统的一个静态的概念模型。

4．业务程序模型（BPM）

BPM 用来描述业务对象的任务和内在流程。它从业务人员的角度来看待业务逻辑和规则。

第 12 章　NUnit 实例入门

NUnit 是为.NET 框架设计的开放源代码单元测试框架。NUnit 支持.NET 框架所支持的各种编程语言，您可以选择自己喜欢的语言编写测试类。下面我们将通过一个翔实的例子，详细介绍如何利用 NUnit 单元测试框架进行应用程序类的单元测试工作。

本文中，我们使用的是 VisualStudio.NET 2005，编程语言是 C#。

12.1 NUnit 下载安装

我们可以在 http://www.nunit.org 网站上下载 NUnit，目前最新的版本是 NUnit 2.4 版本。

NUnit 2.4.8 (Recommended)

win .net 1.1	NUnit-2.4.8-net-1.1.msi	针对.NET1.x 版本，打包成 Windows 安装文件
win .net 2.0	NUnit-2.4.8-net-2.0.msi	针对.NET2.0 版本，打包成 Windows 安装文件
bin .net 1.1	NUnit-2.4.8-net-1.1.zip	针对.NET1.x 版本，未打包文件
bin .net 2.0	NUnit-2.4.8-net-2.0.zip	针对.NET2.0 版本，未打包文件
src	NUnit-2.4.8-src.zip	程序源文件
doc	NUnit-2.4.8-doc.zip	程序说明文件

可以针对用户机器上所用的.NET 版本下载指定的 Msi 或 Zip 文件。下载 MSI 文件后直接双击即可选择安装路径进行安装；下载 Zip 文件后则解压在用户指定目录。两者选一即可。

12.2 创建测试项目

STEP 01 在 VS.NET 开发环境中，单击“文件”菜单上指向“新建”，选择“项目…”。

STEP 02 在“新建项目”对话框中，选择“项目类型”窗格中的“Visual C#项目”，然后选择“模板”中“类库”，在“名称”框中键入“UnitTestExample”。

STEP 03 单击“确定”关闭对话框。

STEP 04 项目创建，并且“Class1.cs”被打开。

12.3 添加 NUnit 引用

STEP 01 在“项目”菜单上，选择“添加引用…”

STEP 02 在“添加引用”对话框中，选择“浏览”页面，寻找 NUnit 安装目录“bin”目录下 nunit.framework.dll。

STEP 03 单击“确定”关闭对话框。可以看到在“解决方案管理器”窗格“引用”目录下增加了“nunit.framework”引用，如图 12-1 所示。

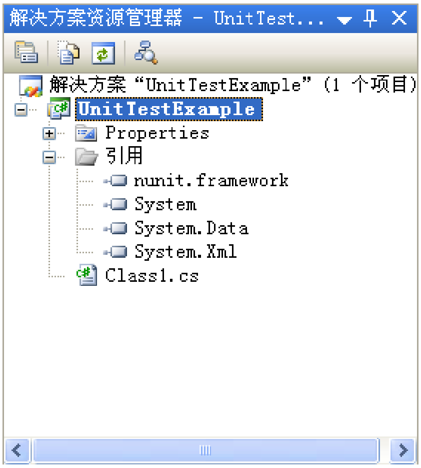


图 12-1　添加 NUnit 引用

STEP 04 在 Class1.cs 前添加引用，即加入“using NUnit.Framework;”语句。注意大小写。

12.4 创建测试用例

STEP 01 把 Class1 修改为 BookTest，并作以下修改：

```
[TestFixture]
public class BookTest
{
  [Test]
  public void Add()
  {
    Book book = new Book();
    book.Add(100);
    Assert.AreEqual (100, book.Amount);
  }
}
```

注意，其中的[TestFixture]和[Test]两个 Attribute 为 NUnit 所规定必须要添加的，这样，测试框架就可以知道哪些类或者方法需要进行测试。

我们在 Add 方法中定义了一个 book 对象，并给他加了 100 本书，然后就可以使用：Assert.AreEqual (100, book.Amount);来测试 book 的 Amount 属性是否确实为 100。

STEP 02 接下来，我们再向 BookTest 中添加一个测试 Sell 的方法。

```
[Test]
public void Sell()
{
  Book book = new Book ();
  book.Add(100);
  book.Sell();
  book.Sell();
  book.Sell();
  Assert.AreEqual (97, book.Amount);
}
```

这里，我们先加了 100 本书之后就一口气卖掉了 3 本，然后看看我们是否还剩下 97 本。

12.5 创建被测试类

在 Class1.cs 中添加 Book 类如下：

```
public class Book
{
  private int amount;
  public int Amount
  {
    get
    {
      return amount;
    }
  }
public void Add(int num)
{
}
public void Sell()
{
}
}
```

注意：这段代码只是为了完成类的结构，方法的实现暂时先空着。

12.6 编译项目

STEP 01 按 F6 键，编译代码。

STEP 02 形成一个 dll 动态连接库文件：UnitTestExample.dll。可以项目文件夹“bin\debug”目录在找到（项目所在的文件夹，可以通过单击“解决方案管理器”，“属性”窗格中，“项目文件夹”属性获得）。

12.7 运行图形测试工具

STEP 01 寻找 NUnit 安装目录“bin”目录下“nunit.exe”，单击运行（如果是以 msi 方式安装的，则会在桌面上创建该文件的快捷方式）。

STEP 02 在 NUnit 图形界面中，在“File”菜单中，选择“Open Project…”，打开我们编译好的 dll 文件。

STEP 03 单击“Run”按钮，出现很醒目的红色，表示测试没有成功。由于前面 Book 类的方法都是空，显然测试无法成功。运行后界面如图 12-2 所示。

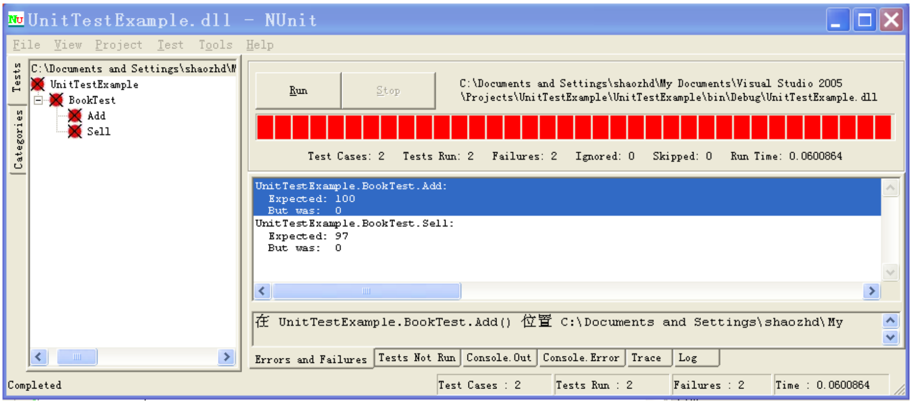


图 12-2 NUnit 运行界面

12.8 修改测试

STEP 01 完成 Book 类的 Add 方法。

```
public void Add(int num)
{
  amount += num;
}
```

STEP 02 保存，重新编译。切换到 NUnit，再单击 Run 按钮，可以看到 Add 方法已经变成绿色了，证明 Add 方法已经成功。

STEP 03 修改 Book 类的 Sell 方法。

```
public void Sell()
{
  amount -= 1;
}
```

STEP 04 再测试，Sell 方法也测试通过了。

12.9 异常测试

STEP 01 编写测试代码。为 BookTest 类添加 ExcpetionTesting 方法：

```
[Test]
[ExpectedException(typeof(Exception))]
public void ExcpetionTesting()
{
    Book book = new Book ();
    book.Add(3);
    book.Sell();
    book.Sell();
    book.Sell();
    book.Sell();
}
```

其中，[ExpectedException(typeof(Exception))]表示我们希望能捕获到发生的异常，如果没有捕获到异常，则表示测试失败。

后面的代码表明添加了三本书，然后卖了四本。显然这样会导致出错。

STEP 02 保存，重新编译。切换到 NUnit，再单击 Run 按钮，可以看到 ExcpetionTesting 方法前为红色，表明测试没有通过。

STEP 03 修改被测试类。在 Book 类中，我们修改一下 Sell 方法，让它变成：

```
public void Sell()
{
  if(amount - 1 < 0)
  throw new Exception("Amount 不能为 0");
  amount -= 1;
}
```

STEP 04 编译，再测试，测试通过！

第 13 章　JUnit 实例入门

JUnit 是一个开源的 Java 单元测试框架，用于编写和运行可重复的测试。在 1997 年，由 Erich Gamma 和 Kent Beck 开发完成。它是用于单元测试框架体系 xUnit 的一个实例（用于 Java 语言）。它包括以下特性：

- 用于测试期望结果的断言（Assertion）
- 用于共享共同测试数据的测试工具
- 用于方便的组织和运行测试的测试套件
- 图形和文本的测试运行器

接下来，我们以一个实例向大家介绍 JUnit 的使用。本文中，我们使用的开发工具是 Eclipse3.4.1，编程语言是 Java。

13.1 JUnit 下载和安装

STEP 01 在 http://www.junit.org 上可以下载到最新版本的 JUnit，本文使用的为 4.5 版本。

STEP 02 下载完以后解压缩到指定的目录下，假设是 JUNIT_HOME，然后将 JUNIT_HOME 下的 junit.jar 包加到你的系统的 CLASSPATH 环境变量中，对于 IDE 环境，对于需要用到的 junit 的项目增加到 lib 中，其设置不同的 IDE 有不同的设置。

13.2 创建测试项目

STEP 01 运行 Eclipse，单击“File 件”菜单上指向“New”，选择“Java Project”。

STEP 02 在“New Java Project”对话框中，在“Project Name”对话框中键入“UnitTestExample”，如图 13-1 所示。

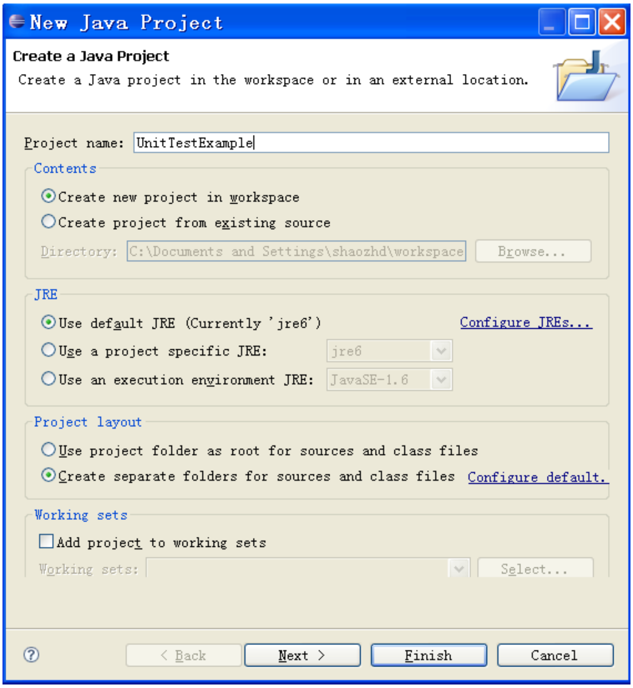


图 13-1 “新建工程”对话框

STEP 03 单击“Finish”按钮，生成该项目。可以看到在主窗体“Package Explore”中生成了“UnitTestExample”项目。

13.3 添加 JUnit 引用

STEP 01 在“Package Explore”窗格选中“UnitTestExample”，单击右键，选择菜单“Build Path\Configure Build Path…”，弹出“Properties for UnitTestExample”对话框，

在右面窗格中选择“Libraries”属性页，如图 13-2 所示。

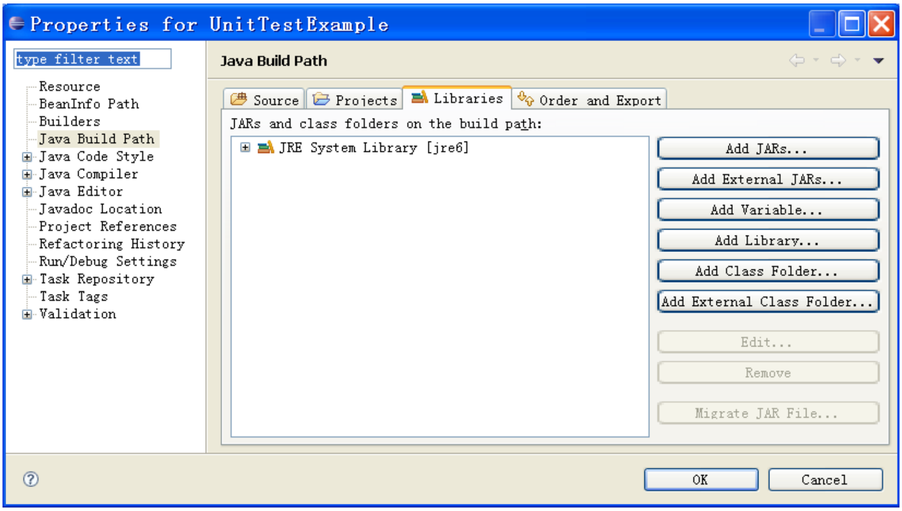


图 13-2 “UnitTestExample”项目属性

STEP 02 单击“Add Library…”按钮，弹出“Add Library”对话框，可以看到 JUnit 运行包已经内置在 Eclipse 中，如图 13-3 所示。

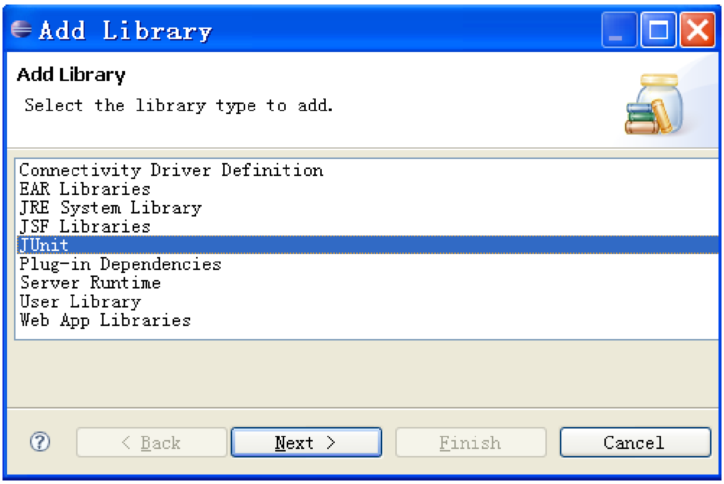


图 13-3 “Add Library”对话框

STEP 03 单击“Next”按钮，弹出“JUnit Labrary”对话框，可以选择 JUnit 版本，选择“JUnit 4”，单击“Finish”按钮加入该包，如图 14-4 所示。

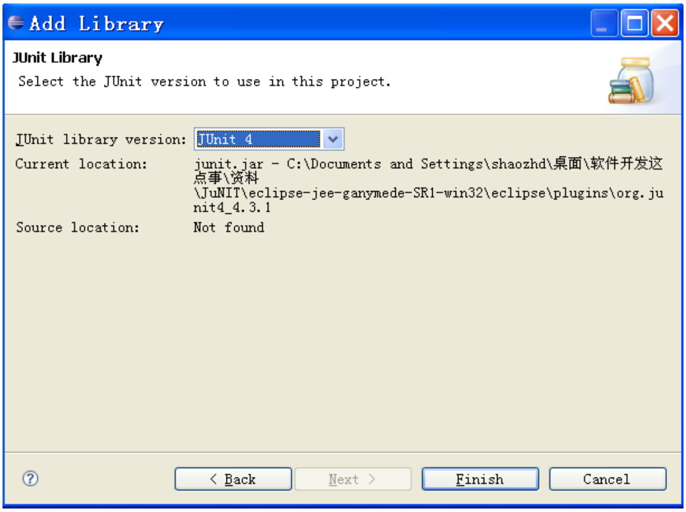


图 13-4　JUnit 库文件选择界面

STEP 04 依次单击“Finish\OK”按钮，返回主窗体，可以看到在“Package Explore”中已经添加了“JUnit4 的引用”。

13.4 创建被测试类

STEP 01 在“Package Explore”窗格中选中 Src 目录，右键单击，在出现的菜单中选择“New/Class”，弹出“New Java Class”对话框，在“Name：”文本框中输入“Book”，单击“Finish”按钮。

STEP 02 在 Book.java 中输入以下代码：

```
public class Book {
    private int amount;
    public int getAmount() {
          return amount;
    }
    public void setAmount(int amount) {
          this.amount = amount;
    }
    public void Add(int num){
         amount += num;
    }
    public void Sell(){
```

```
            amount -= 1;
        }
    }
```

13.5 创建测试用例

STEP 01 在“Package Explore”窗格中选中“Src\default pakage”目录，右键单击，在出现的菜单中选择“New/Class”，弹出“New Java Class”对话框，在“Name：”文本框中输入“BookTest”，单击“Finish”按钮。

STEP 02 在 BookTest.java 中输入以下代码：

```
import junit.framework.TestCase;
public class BookTest extends TestCase{
    public void testAdd()
      {
        Book book = new Book();
        book.Add(100);
        assertEquals(100, book.getAmount());
      }
    public void testSell()
    {
    Book book = new Book ();
    book.Add(100);
    book.Sell();
    book.Sell();
    book.Sell();
    assertEquals(97, book.getAmount());
    }
```

注意：

- 测试用例要从 TestCase 继承。
- 要在测试方法前加“test”字样，这样，测试框架就可以知道哪些方法需要进行测试。

我们在 testAdd 方法中定义了一个 book 对象，并给他加了 100 本书，然后就可以使用：assertEquals 来测试 book 的 amount 值是否确实为 100。

在 testSell 方法中，先加了 100 本书，然后卖掉了 3 本，然后看看我们是否还剩下 97 本书。

13.6 创建测试单元

STEP 01 在“Package Explore”窗格中选中“Src\default pakage”目录，右键单击，在出现的菜单中选择“New/Class”，弹出“New Java Class”对话框，在“Name：”文本框中输入“Main”，并把“pulic static void main(String[] args)”前面的复选框打钩，如图 13-5 所示。

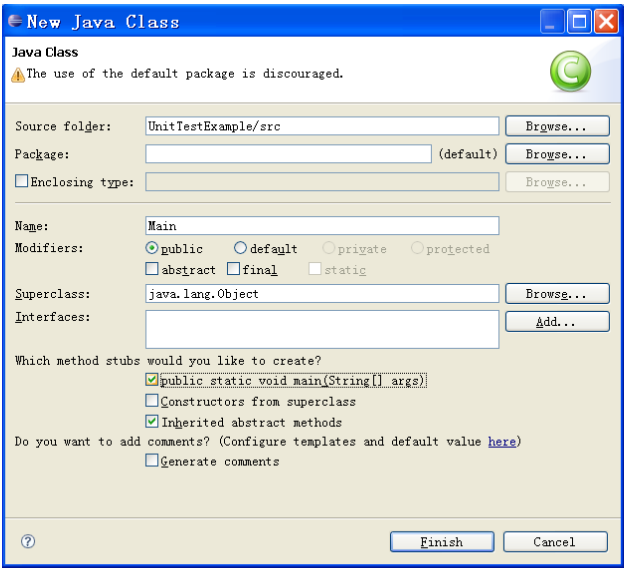


图 13-5 “添加新 Java 类”对话框

STEP 02 单击“Finish”按钮，添加该类。

STEP 03 在 Main.java 文件前添加以下代码，引用 JUnit 包：

```
import junit.framework.TestSuite;
import junit.framework.Test;
import junit.textui.TestRunner;
```

STEP 04 修改 Main 类：

```
public class Main extends TestSuite {
    public static Test suite() {
        TestSuite suite = new TestSuite("TestSuite Test");
        suite.addTestSuite(BookTest.class);
        return suite;
```

```
        }
        public static void main(String args[]){
            TestRunner.run(suite());
        }
}
```

13.7 运行测试

STEP 01 在“Package Explore”窗格中右键单击“Main.java”，在弹出的菜单中选择“Run As”下“2 Java Application”，可以看到输出窗口中有如下显示，如图 13-6 所示。

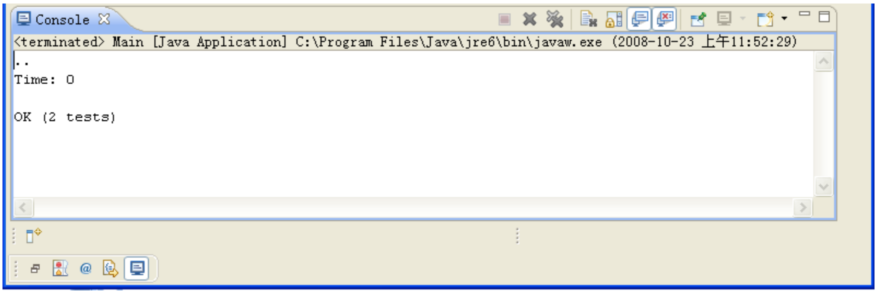


图 13-6 输出窗口

STEP 02 在“Package Explore”窗格中右键单击“Main.java”，在弹出的菜单中选择“Run As”下“3 JUnit Test”，会弹出 JUnit 图形界面，如图 13-7 所示。

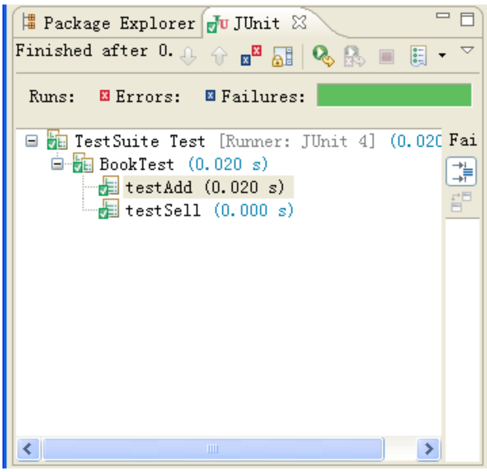


图 13-7 JUnit 图形界面

13.8 JUnit 结构说明

1．软件结构

JUnit 共有 7 个包，核心的包就是 junit.framework 和 junit.runner。Framework 包负责整个测试对象的构架，Runner 负责测试驱动。

2．类结构

JUnit 有 4 个重要的类：TestSuite、TestCase、TestResult、TestRunner。前三个类属于 Framework 包，后一个类在不同的测试环境下是不同的。例如，当使用文本测试环境时，所用的是 junit.textui.TestRunner。各个类说明如下：

- TestResult：负责收集 TestCase 所执行的结果，它将结果分为两类，客户可预测的 Failure 和没有预测的 Error。同时负责将测试结果转发到 TestListener（该接口由 TestRunner 继承）处理。
- TestRunner：客户对象调用的起点，负责对整个测试流程的跟踪。能够显示返回的测试结果，并且报告测试的进度。
- TestSuite：负责包装和运行所有的 TestCase。
- TestCase：客户测试类所要继承的类，负责测试时对客户类进行初始化，以及测试方法调用。

另外，还有两个重要的接口：Test 和 TestListener。

- Test：包含 2 个方法：run()和 countTestCases()，它是对测试动作特征的提取。
- TestListener：包含 4 个方法：addError()、addFailure()、startTest()和 endTest()，它是对测试结果的处理以及测试驱动过程的动作特征的提取。

3．测试流程

典型的使用 JUnit 的方法就是继承 TestCase 类，然后重载它的一些重要方法：setUp()、teardown()、runTest()（这些都是可选的），最后将这些客户对象组装到一个 TestSuite 对象中，交由 junit.textui.TestRunner.run（案例集）驱动。

第 14 章　InstallShield 实例入门

InstallShield 是制作安装包最常用的工具，它已经成为了事实上的安装标准，是程序员的一个非常好的助手工具。

下面我们将用一个实例来说明使用 InstallShield 来制作安装程序。我们使用的安装工具是 InstallShield2009。部署的应用程序是一个 Web 网站（只提供了一个 Index.html 页面，位于图书配盘 Demo\InstallShied\TestWebSite 目录，读者也可以用其他的页面代替），所使用的数据库是 SQL Server，在安装的过程中应该自动创建数据库和表。

该实例位于光盘的“DEMO\InstallShield\”目录下，双击 TestProject.ism 文件可以用 InstallShield 打开。

14.1 创建安装项目

STEP 01 启动 InstallShield。

STEP 02 在“File”菜单上，单击“New…”。

STEP 03 在“New Project”对话框中，选择“Windows Installer”页面中的“Web Project”，在“Project Name”中输入“TestProject”。可以看到程序的默认存放路径为“C:\InstallShield 2009 Projects”。

STEP 04 单击“OK”按钮。可以看到如图 14-1 所示的界面。

在界面最下方有 9 项设置可以供用户选择，分别为“Application Information”（程序信息）、“Installation Requirements”（程序运行条件）、“Installation Architecture”（安装架构）、“Application Files”（要发布的程序文件）、“Application Shortcuts”（程

序快捷键）、“Application Registry”（程序注册表）、“Installation Interview”（安装程序界面）、“Installation Localization”（安装程序语言）、“Build Installation”（生成安装程序）。使用这些安装程序制作向导页面，我们可以轻松制作安装程序。

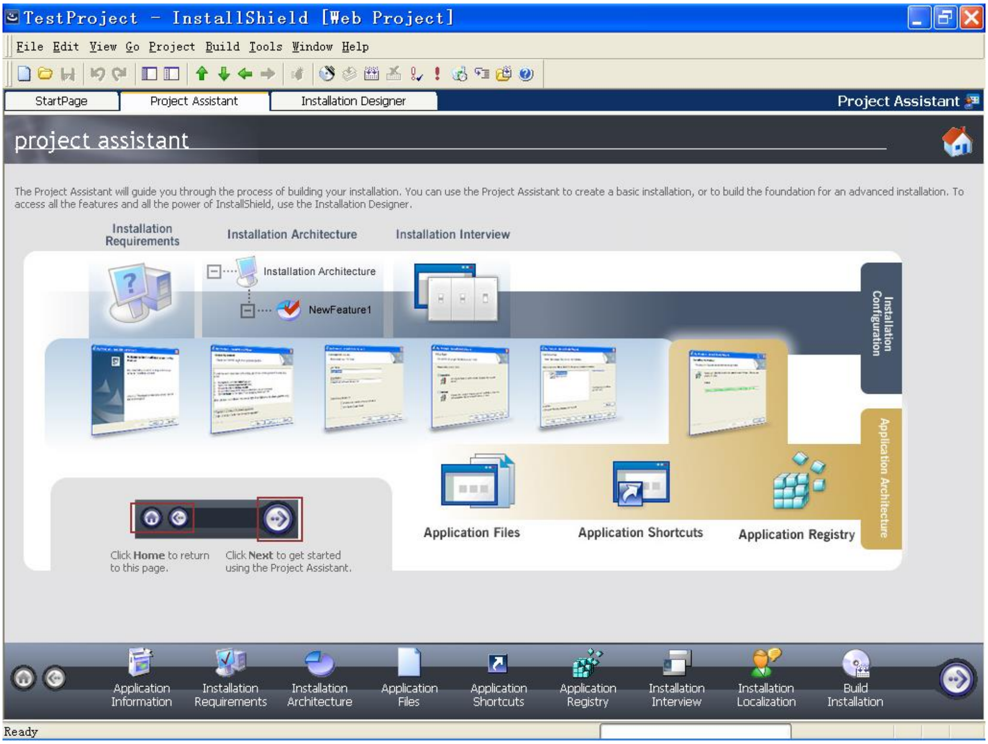


图 14-1　InstallShield 2009 界面

14.2 安装程序语言设计

STEP 01 为了使程序支持多语言安装，我们首先要设置安装程序语言。在“Project Assistant”页面，单击左下方的“Installation Locallization”图标，切换到语言设置页面。

STEP 02 在该页面的下拉框中，选择“All String Data”，在下面的语言列表中，我们选择“Chinese（Simplified）”和“Chinese（Traditional）”，使得安装程序支持中文简体和中文繁体两个版本。

14.3 应用程序信息设置

STEP 01 在“Project Assistant”页面，单击左下方的“Application Information”图标，切换到该页面。

STEP 02 在“Specify your company name：”文本框中输入“MyCompany”；在“Specify your application name：”文本框中输入：“Electronic Commerce System”；下面两个文本框分别为程序版本号和公司网站名称。

14.4 设置程序运行条件

STEP 01 在“Project Assistant”页面，单击左下方的“Installation Requirements”图标，切换到该页面。

STEP 02 把“Does your application require any specify operating”下方的“Yes”选中。表明该应用程序只运行于 Windows 2000 以后的平台。

14.5 设定安装架构

STEP 01 在“Project Assistant”页面，单击左下方的“Installation Architecture”图标，切换到该页面。

STEP 02 选择页面“Do you want to customize your Installation”下面的“Yes”单选按钮。

STEP 03 单击页面中“Installation Architecture”节点，再单击下方“New”按钮，把新出现的“New_Feature1”命名为“Help File”。页面如图 14-2 所示。

14.6 添加要发布的网站文件

STEP 01 在“Project Assistant”页面，单击左下方的“Application Files”图标，切换到该页面。

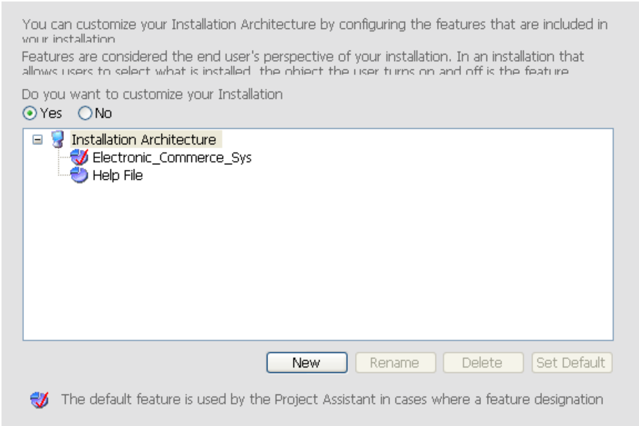


图 14-2 “Application Files”页面

STEP 02 在下拉框中，选择“Electronic_Commerce_Sys”，选择左边列表框中的“[IISROOTFOLDER]”，单击屏幕右下方的“Add Folders”按钮，在弹出的对话框中选定本书附带光盘“DEMO\Chapter8\Insallshield\TestWebSite”目录。单击“确定”按钮，添加该目录。系统会弹出一个对话框，询问是否要建立动态链接（选择“Yes”即建立连接，否则复制文件。对于安装文件特别多的情况，可以选择“是”，此处我们选“No”），单击“No”。可以看到在[IISROOTFOLDER]添加了“TestWebSite”目录。

STEP 03 选择“TestWebsite”目录，右键单击右边列表中的“help.html”，选择“Delete”删除该文件。

STEP 04 在下拉框中，选择“Help File”，选择左边列表框中的“[IISROOTFOLDER]”，单击右键，在弹出的对话框中，选择“New Folder”。把“New Folder 1”重命名为“TestWebSite”（注意大小写必须相同），并选中它。

STEP 05 单击屏幕右下方的“Add Files”按钮，在弹出的“打开”对话框中选定本书附带光盘“DEMO\Chapter8\Insallshield\TestWebSite”目录下的“Help.html”文件，单击“打开”按钮。

14.7 添加程序快捷键

STEP 01 在“Project Assistant”页面，单击左下方的“Application Shortcuts”图标，切换到该页面。

STEP 02 单击屏幕中的“New”按钮，弹出“Browse for a destination file”对话框，把“Files of”下拉框中选择为“All Files（*.*）”，双击“[IISROOTFOLDER]\TestWebsite”，选择“Index.html”，单击“Open”按钮返回。可以看到在页面列表中生成了一个名为“Launch index.html”的快捷方式。

STEP 03 选中该快捷方式，单击下面的“Rename”按钮，修改其名称为“Electronic Commerce System1.0”。

STEP 04 把右面的“Create shortcut on desktop”复选框打钩。

14.8 添加注册表修改

STEP 01 在“Project Assistant”页面，单击左下方的“Application Registry”图标，切换到该页面。

STEP 02 单击页面上方的“Yes”按钮，允许注册表修改。

STEP 03 单击列表中的“HK_CURRENT_USER”，右键单击该项，在弹出的菜单项中选择“New—Key”，把新生成的键“New Key #1”重命名为“MyCompany”。

STEP 04 右键单击“MyCompany”，在弹出的菜单项中选择“New—String Value”，把新生成的值“New Value #1”重命名为“CompanyName”。

STEP 05 双击“CompanyName”，弹出的“Edit Data：”对话框，修改“Value Data”文本框值为“This is my company”。设置后界面如图14-3所示。

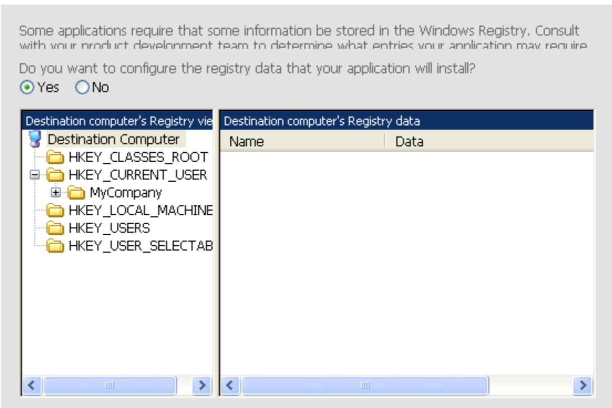


图14-3 添加注册表键值

14.9 安装界面设置

STEP 01 在“Project Assistant”页面，单击左下方的“Installation Interview”图标，切换到该页面。

STEP 02 单击“Do you want your users to be able to selectively install only certain parts of your application？”下的“Yes”单选框。选择后界面如图 14-4 所示。

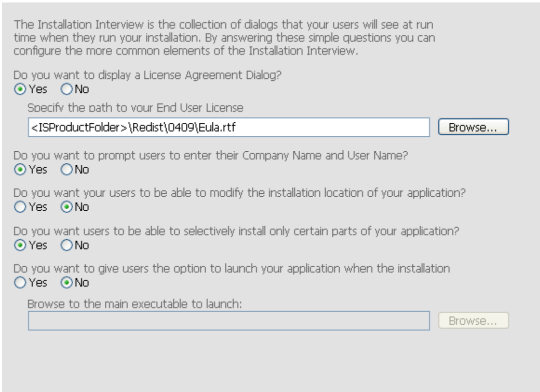


图 14-4 “Installation Interview”界面

14.10 多语言字符串修改

STEP 01 选择“Installation Designer”页面，选择最左边列表“Gerneral Information”，选择中间列表中的“String Tables”，可以看到下面有两个语言版本。

STEP 02 把两个语言版本的“ID_STRING1”都设置为“http://www.MyCompany.com”；把“IDS_SHORTCUT_DISPLAY_NAME5”设置为“Electronic Commerce System1.0”；把“IDS_FEATURE_DISPLAY_NAME3”设置为“Help File”。

14.11 生成安装

STEP 01 选择“Project Assistant”页面，单击左下方的“Build Installation”图标，

切换到生成页面。

STEP 02 有四种生成方式可以选择：单独 EXE 执行文件、网络安装程序、CD-ROM 方式、单独 MSI 包。我们只选择第一种方式。在“Single Executable”前的复选框打钩。

STEP 03 单击屏幕下方的“Build Installations”按钮。开始生产安装文件。

14.12 运行安装文件

本例中我们以 SQL Server 2000 自带的 NorthWind 数据库为内容，创建一个名称为“MyDB”的数据库。实际项目开发是可以替换为所需要的数据库，也可以自己编写 Script 脚本替换程序中的脚本。

STEP 01 在以下目录可以找到生成的 EXE 版本安装文件“C:\InstallShield 2009 Projects\TestProject\PROJECT_ASSISTANT\SINGLE_EXE_IMAGE\DiskImages\DISK1”。

STEP 02 单击“Setup.exe”即可运行安装程序。

STEP 03 可以看到，安装文件有两个语言可以选择：“中文简体”和“中文繁体”，如图 14-5 所示。

STEP 04 安装类型有“完整安装”和“自定义”两种，如图 14-6 所示。“自定义”安装中可以选择主文件和帮助文件，如图 14-7 所示。

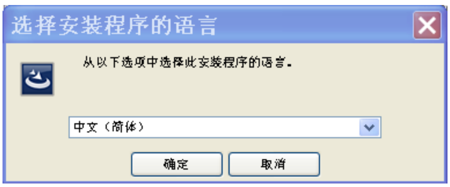


图 14-5 “安装语言选择”对话框



图 14-6 “安装类型选择”对话框

STEP 05 安装完成后会在 Windows“开始”菜单和桌面都创建相应的快捷方式。

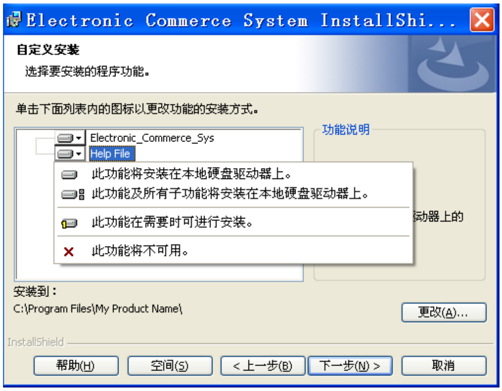


图 14-7 “自定义安装”对话框

STEP 06 安装程序会在计算机注册表中创建我们前面指定的项。注意，由于安装程序要访问系统注册表，请确认有足够权限，如图 14-8 所示。

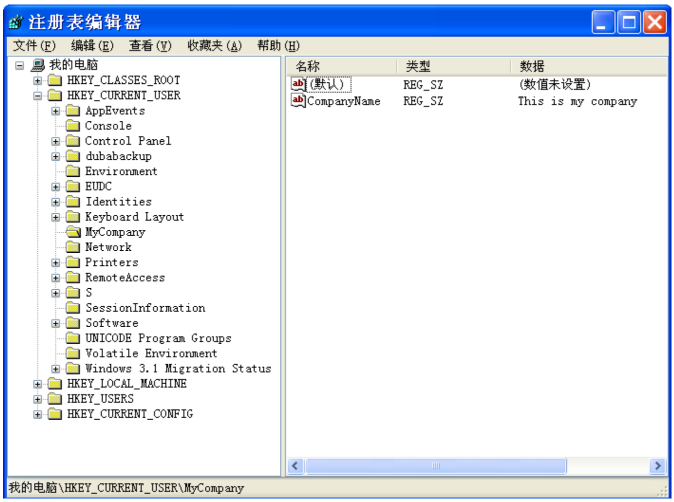


图 14-8 系统注册表

14.13 添加数据库支持

STEP 01 进入“Installation Designer”页面，选择左边列表中“Server Configuration”下的“SQL Scripts”。

STEP 02 右键单击中间列表中的“SQL Scripts”节点，在出现菜单中选择“New SQL

Connection”，把新生成的“NewSQLConnection1”重命名为“TestConnection”。

STEP 03 选择页面最右边面板中的“Requirements”属性页面，单击“Database Servers”下的“Microsoft SQL Server”，单击下方的“Add”按钮。出现数据库选择对话框。在出现的对话框的下拉列表中，选择“SQL Server 6.5 RTM（Release To Manufacturing）”。表明我们要求目标机器上必须安装 SQLServer 6.5 以上版本。

STEP 04 选择页面最右边面板中的“Gerneral”属性页面，单击下面的“Database Import Wizard”链接，启动向导。

STEP 05 在向导中，首先指定所要打包的数据库所在的服务器和登录名。我们分别选择“（local）”和“sa”，如图 14-9 所示。

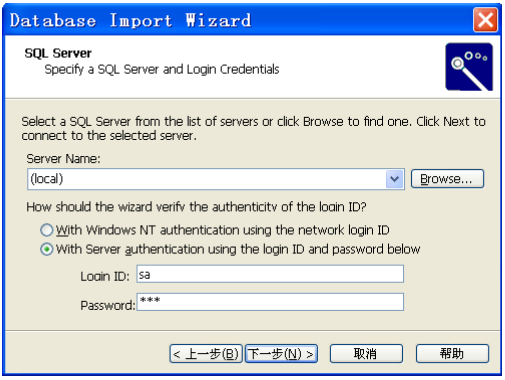


图 14-9 “数据库服务器信息设置”对话框

STEP 06 接下来选择“Northwind”数据库，并设置脚本名称为“TestScript”，如图 14-10 所示。

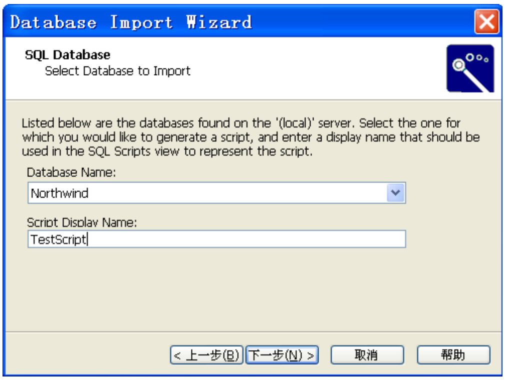


图 14-10 “数据库选择”对话框

STEP 07 单击“下一步”按钮，进入“选择数据库表”对话框。我们选择所有表，如图 14-11 所示。

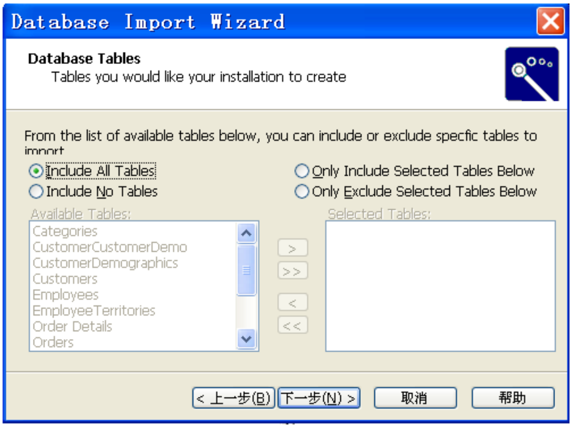


图 14-11 “数据表选择”对话框

STEP 08 单击“下一步”按钮，进入“选择创建数据库对象”对话框，使用默认选择，如图 14-12 所示。

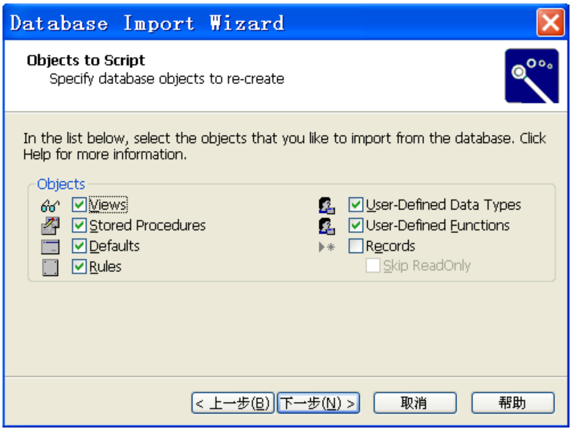


图 14-12 “数据库对象选择”对话框

STEP 09 单击“下一步”按钮，进入“脚本创建选项”对话框，做如下选择，如图 14-13 所示。

STEP 10 单击“下一步”按钮，进入“高级选项”对话框。做如下选择，如图 14-14 所示。



图 14-13 “脚本创建选项”对话框

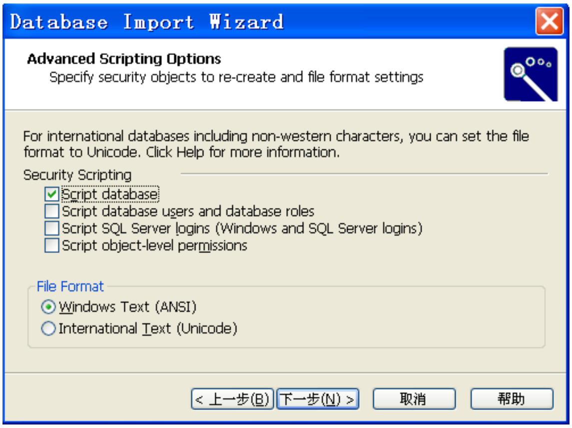


图 14-14 “高级选项”对话框

STEP 11 单击“下一步”按钮，出现最后的对话框，单击“完成”按钮。

STEP 12 在中间面板，选中“TestScript”节点；在右面面板，选中“Script”属性页，可以看到产生的脚本。

STEP 13 把脚本中“northwnd.mdf”修改为“MyDB.mdf”，把“Northwind_log.LDF”修改为“MyDB_log.LDF”。在“CREATE DATABASE [Northwind]…”后一行的“GO”语句后添加如下两行语句：

```
Use    [Northwind]
GO
```

STEP 14 选择“Text Replacement”属性页，单击下方“Add…”按钮，弹出“Find

& Replace”对话框，在“Find What：”后输入“Northwind”，在“Replace With：”后输入“MyDB”。单击“OK”按钮返回。

STEP 15 按“F7”键重新生成安装程序。按“Ctrl+F5”组合键运行安装程序，可以发现比原来多了一个对话框，需要安装用户选择数据库服务器和输入用户验证，如图 14-15 所示。



图 14-15　安装界面：数据库服务器选择

STEP 16 定制“SQL 登录”对话框。上面的对话框可以在“Installation Designer→User Interface→Dialogs→All Dialogs”下找到，名字为 SQLLogin。由于我们不希望用户选择数据库，而直接使用 MyDB 数据库，所以把以上的“数据库编录的名称”标签和下方的文本框、“浏览…”按钮的“Visable”属性设置为 False。注意，要把两个语言版本都进行修改。修改如图 14-16 所示。

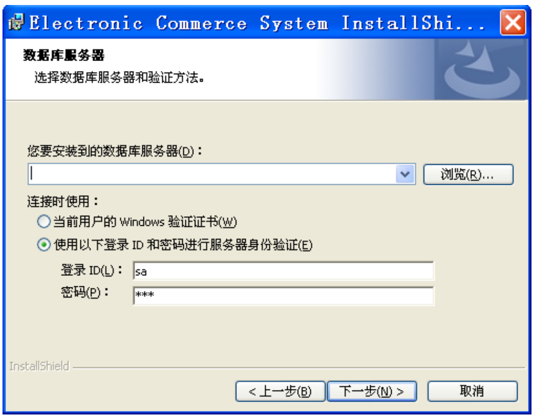


图 14-16　安装界面：数据库服务器选择

STEP 17 安装程序完成后，打开数据库，发现创建 MyDB 数据库成功！

14.14 直接附加数据库

STEP 01 首先把“MyDB2.mdf”和“MyDB2.ldf 文件添加到安装文件中去。具体可以参照第 6 步（在安装盘中可以找到该文件，也可以替换成实际项目数据库文件）。

STEP 02 和上述同样的方法，在“Installation Designer→Server Configuration→SQL scripts”中创建一个新的“SQL Connection”，然后再在该连接下创建一个“TestScript2”，编辑 Script，在脚本框中输入：

```
-- =============================================
-- Create database for attach
-- =============================================
CREATE DATABASE MyDB2
ON PRIMARY (FILENAME = 'c:\dir\MyDB2.mdf')
FOR ATTACH
GO
```

STEP 03 接着在 Text Replacement 中添加一个替换将“c:\dir”替换为“[IISROOTFOLDER]\ TestWebSite\”。

STEP 04 重新编译生成，运行安装程序后，打开 SQL Server，发现创建 MyDB2 成功！

14.15 程序卸载

可以在 Windows 系统“控制面板→添加或删除程序”中，找到已经安装程序，并进行卸载。

第 15 章　“VisualStudio.NET 部署应用程序”实例入门

VisualStudio.NET 是微软出品的开发工具，它继承了微软一贯的用户友好的做法，使用简便而又功能强大。以往的 Visual Studio 只能用于程序开发，从 VisualStudio.NET 2002 版本开始，它也成为了.NET 程序安装部署的重要工具。

下面我们用一个实例来详细说明，如何使用 Visual Studio.NET 来部署应用程序。我们使用的安装工具是 Visual Studio.NET 2005，编程语言是 C#，部署的应用程序是 Web 应用程序，部署过程中需要在 SQLServer 中创建数据库表和初始数据。

该实例位于光盘“Demo\VisualStudio.NET\ WebDeployment”目录下。

15.1 创建要发布 ASP.NET Web 应用程序

STEP 01 运行 VS.NET 2005。在“文件”菜单上，单击“新建”，选择“项目……”。

STEP 02 在“新建项目”对话框上，选择“项目类型”窗格中的“其他项目类型”—“Visual Studio 解决方案”，在“名称”中输入“WebDeployment”。

STEP 03 在“文件”菜单上指向“添加”，然后选择“新建网站……”。

STEP 04 在“添加新网站”对话框上，选择“模板”窗格中的“ASP.NET 网站”，单击“确定”按钮，创建 ASP.NET 网站。

STEP 05 在 Default.aspx 页面上添加一个 Label 控件，并设置其 Text 属性为“欢迎使用学校投诉系统 1.0！”（或其他语句）。

15.2 创建安装程序类

STEP 01 在“文件”菜单上单击“添加”→“新建项目”。

STEP 02 在“添加新项目”对话框中，选择“项目类型”窗格中的“Visual C#项目”，然后选择“模板”窗格中的“类库”。在“名称”框中键入 DBCustomAction。单击“确定”按钮关闭对话框。

STEP 03 在“项目”菜单上单击“添加新项”。

STEP 04 在“添加新项”对话框中选择“安装程序类”。在“名称”框中键入 DBCustomAction.cs。

15.3 创建包含 SQL 语句的文本文件以创建数据库

STEP 01 在解决方案资源管理器中，选择 DBCustomAction 项目。在“项目”菜单上选择“添加新项”。

STEP 02 在“添加新项”对话框中，选择“文本文件”。在“名称”框中，键入 sql.txt（必须小写）。

STEP 03 将以下内容添加到 sql.txt 文件中（文件内容能够可以根据实际需要修改和替换）。可以在光盘“Demo\VisualStudio.net\WebDemployment\DBCustomAction”目录下找到该文件。

```
CREATE TABLE [dbo].[request] (
    [requestid] [int] IDENTITY (1, 1) NOT NULL ,
    [studentid] [int] NULL ,
    [studentname] [nvarchar] (25) COLLATE Chinese_PRC_CI_AS NULL ,
    [class] [nvarchar] (25) COLLATE Chinese_PRC_CI_AS NULL ,
    [phone] [nvarchar] (50) COLLATE Chinese_PRC_CI_AS NULL ,
    [email] [nvarchar] (80) COLLATE Chinese_PRC_CI_AS NULL ,
    [title] [nvarchar] (100) COLLATE Chinese_PRC_CI_AS NULL ,
    [content] [ntext] COLLATE Chinese_PRC_CI_AS NULL ,
    [urgency] [nvarchar] (15) COLLATE Chinese_PRC_CI_AS NULL ,
    [important] [nvarchar] (15) COLLATE Chinese_PRC_CI_AS NULL ,
    [createtime] [datetime] NULL ,
    [lasttime] [datetime] NULL ,
```

```
    [processstate] [nvarchar] (20) COLLATE Chinese_PRC_CI_AS NULL ,
    [username] [nvarchar] (25) COLLATE Chinese_PRC_CI_AS NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]


CREATE TABLE [dbo].[tblStudentInfo] (
    [StudentID] [int] IDENTITY (100000, 1) NOT NULL ,
    [EnterDate] [datetime] NULL ,
    [Name] [nvarchar] (40) COLLATE Chinese_PRC_CI_AS NULL ,
    [Pwd] [nvarchar] (20) COLLATE Chinese_PRC_CI_AS NULL ,
    [Sex] [nvarchar] (10) COLLATE Chinese_PRC_CI_AS NULL ,
    [BirthDate] [datetime] NULL ,
    [PersonalCode] [nvarchar] (30) COLLATE Chinese_PRC_CI_AS NULL ,
    [ListenCode] [nvarchar] (10) COLLATE Chinese_PRC_CI_AS NULL ,
    [HomeTel] [nvarchar] (24) COLLATE Chinese_PRC_CI_AS NULL ,
    [OfficalTel] [nvarchar] (24) COLLATE Chinese_PRC_CI_AS NULL ,
    [Mobile] [nvarchar] (24) COLLATE Chinese_PRC_CI_AS NULL ,
    [BBCall] [nvarchar] (30) COLLATE Chinese_PRC_CI_AS NULL ,
    [Email] [nvarchar] (40) COLLATE Chinese_PRC_CI_AS NULL ,
    [Homepage] [nvarchar] (40) COLLATE Chinese_PRC_CI_AS NULL ,
    [Jiguan] [nvarchar] (30) COLLATE Chinese_PRC_CI_AS NULL ,
    [Hukou] [nvarchar] (30) COLLATE Chinese_PRC_CI_AS NULL ,
    [Post] [nvarchar] (100) COLLATE Chinese_PRC_CI_AS NULL ,
    [PostalCode] [nvarchar] (10) COLLATE Chinese_PRC_CI_AS NULL ,
    [HomeAddress] [nvarchar] (100) COLLATE Chinese_PRC_CI_AS NULL ,
    [Address] [nvarchar] (100) COLLATE Chinese_PRC_CI_AS NULL ,
    [Uni] [nvarchar] (40) COLLATE Chinese_PRC_CI_AS NULL ,
    [Spec] [nvarchar] (40) COLLATE Chinese_PRC_CI_AS NULL ,
    [GradTime] [nvarchar] (10) COLLATE Chinese_PRC_CI_AS NULL ,
    [Edu] [nvarchar] (40) COLLATE Chinese_PRC_CI_AS NULL ,
    [EngLevel] [nvarchar] (50) COLLATE Chinese_PRC_CI_AS NULL ,
    [StrongSuit] [nvarchar] (100) COLLATE Chinese_PRC_CI_AS NULL ,
    [Hobby] [nvarchar] (200) COLLATE Chinese_PRC_CI_AS NULL ,
    [EduList1] [nvarchar] (200) COLLATE Chinese_PRC_CI_AS NULL ,
    [EduList2] [nvarchar] (200) COLLATE Chinese_PRC_CI_AS NULL ,
    [EduList3] [nvarchar] (200) COLLATE Chinese_PRC_CI_AS NULL ,
    [EduList4] [nvarchar] (200) COLLATE Chinese_PRC_CI_AS NULL ,
    [EduList5] [nvarchar] (200) COLLATE Chinese_PRC_CI_AS NULL ,
    [WorkList1] [nvarchar] (200) COLLATE Chinese_PRC_CI_AS NULL ,
    [WorkList2] [nvarchar] (200) COLLATE Chinese_PRC_CI_AS NULL ,
    [WorkList3] [nvarchar] (200) COLLATE Chinese_PRC_CI_AS NULL ,
    [WorkList4] [nvarchar] (200) COLLATE Chinese_PRC_CI_AS NULL ,
    [WorkList5] [nvarchar] (200) COLLATE Chinese_PRC_CI_AS NULL ,
    [EduTgt] [nvarchar] (50) COLLATE Chinese_PRC_CI_AS NULL ,
    [ToPra] [nchar] (1) COLLATE Chinese_PRC_CI_AS NULL ,
```

```
        [InfoSrc] [nvarchar] (10) COLLATE Chinese_PRC_CI_AS NULL ,
        [RecWork] [nchar] (1) COLLATE Chinese_PRC_CI_AS NULL ,
        [ClassCode] [varchar] (50) COLLATE Chinese_PRC_CI_AS NULL
    ) ON [PRIMARY]


    CREATE TABLE [dbo].[tblUser] (
        [id] [int] IDENTITY (1, 1) NOT NULL ,
        [UserName] [nchar] (20) COLLATE Chinese_PRC_CI_AS NULL ,
        [ShowName] [nvarchar] (255) COLLATE Chinese_PRC_CI_AS NULL ,
        [Password] [nvarchar] (255) COLLATE Chinese_PRC_CI_AS NULL ,
        [EnterTime] [datetime] NULL ,
        [Phone] [int] NULL ,
        [Email] [varchar] (50) COLLATE Chinese_PRC_CI_AS NULL ,
        [Position] [varchar] (50) COLLATE Chinese_PRC_CI_AS NULL ,
        [Quanxian] [bit] NULL
    ) ON [PRIMARY]
    Insert tblStudentInfo (name,pwd,ClassCode) values('张三','111','IJ04')
    Insert tblUser (Username,showname,password,position,quanxian) values
('Admin','管理员','111','管理员',1)
    Insert tblUser (Username,showname,password,position,quanxian) values
('lisi','李四','111','连锁人员',0)
    Insert tblUser (Username,showname,password,position,quanxian) values
('shaozhd','邵志东','111','教师',0)
```

STEP 04 在“解决方案资源管理器”中选择 SQL.txt。在“属性”窗口中，将“生成操作”属性设置为“嵌入的资源”。

15.4 将代码添加到安装程序类中以阅读文本文件

STEP 01 在“解决方案资源管理器”中，选择“DBCustomAction.cs”，在“视图”菜单上选择“代码”。

STEP 02 在模块的顶部添加下面的 Using 语句：

```
    using System.IO;
    using System.Data.SqlClient;
    using System.Reflection;
```

STEP 03 将下面的代码添加到 DBCustomAction 类：

```
    public override void Install(System.Collections.IDictionary stateSaver)
           {
```

```
                base.Install(stateSaver);
                if (!InstallDB("RequestSys"))//RequestSys 为数据库名称
                      base.Uninstall(stateSaver);

            }
            private string GetSql(string strName)
            {
                try
                {
                    //' Get the current assembly.
                    Assembly Asm = Assembly.GetExecutingAssembly();
                    // Resources are named using a fully qualified name

                    Stream strm = Asm.GetManifestResourceStream(Asm.GetName().
Name + "." + strName);

                    //Read the contents of the embedded file.
                    StreamReader reader = new StreamReader(strm);//,System.
Text.Encoding.Unicode);


                    return reader.ReadToEnd();
                }
                catch(Exception e)
                {
                    throw e;
                }

            }
            private void ExecuteSql(string DatabaseName, string Sql)
            {
                //读取参数
                string strSever = this.Context.Parameters["server"];
                string strUser = this.Context.Parameters["user"];
                string strPass = this.Context.Parameters["pwd"];

                System.Text.StringBuilder s = new System.Text.StringBuilder();
                s.Append("user id =");
                s.Append(strUser);
                s.Append(";password=");
                s.Append(strPass);
                s.Append(";database=master;server=");
                s.Append(strSever);

                SqlConnection sqlConnection1 = new SqlConnection(s.ToString());
```

```
            SqlCommand Command = new SqlCommand(Sql, sqlConnection1);
            Command.Connection.Open();
            Command.Connection.ChangeDatabase(DatabaseName);

            try
            {
                Command.ExecuteNonQuery();
            }
            catch (Exception e)
            {
                throw e;
            }
            finally
            {
                // Finally, blocks are a great way to ensure that the
connection
                Command.Connection.Close();
            }

        }
        protected bool InstallDB(string strDBName)
        {
            try
            {
                //Create the database.
                ExecuteSql("master", "CREATE DATABASE " + strDBName);
                // Create the tables.
                ExecuteSql(strDBName, GetSql("sql.txt"));
                return true;
            }
            catch
            {
                return false;
            }
        }
```

STEP 04 在“生成”菜单上，选择“生成 DBCustomAction”。

15.5 创建部署项目

STEP 01 在“文件”菜单中单击“添加”，然后单击“新建项目”。

STEP 02 在“添加新项目”对话框中，打开“其他项目类型”节点，在“项目类

型”窗格中选择“安装和部署项目”。然后，在“模板”窗格中选择“Web 安装项目”，在“名称”框中键入 WebSetupTest。

STEP 03 单击“确定”按钮。项目被添加到解决方案资源管理器中，并且文件系统编辑器打开。

STEP 04 在“属性”窗口中，选择“ProductName”属性并键入“学校投诉系统 1.0”。

STEP 05 在“文件系统编辑器”中，选择“应用程序文件夹”。在“操作”菜单上单击“添加”按钮，然后单击“项目输出”按钮。

STEP 06 在“添加项目输出组”对话框中，为“DBCustomAction”项目选择“主输出”。

15.6 将主程序项目的输出添加到部署项目中

STEP 01 在“文件系统编辑器”中，选择“应用程序文件夹”。在“操作”菜单上，指向“添加”，然后选择“项目输出”。

STEP 02 在“添加项目输出组”对话框中，选择“项目”下拉列表中第一步创建的 Web 应用程序”。

STEP 03 单击“确定”按钮关闭对话框。

15.7 创建自定义安装对话框

STEP 01 在解决方案资源管理器中选择“WebSetupTest”项目。在“视图”菜单上指向“编辑器”，然后选择“用户界面”。

STEP 02 在用户界面编辑器中，选择“安装”下的“启动”节点。在“操作”菜单上，选择“添加对话框”。

STEP 03 在“添加对话框”对话框中，选择“许可协议”对话框，然后单击“确定”按钮关闭对话框。

STEP 04 在“添加对话框”对话框中，选择“文本框（A）”对话框，然后单击“确定”按钮关闭对话框。

STEP 05 在“操作”菜单上，选择“上移”。重复此步骤，直到“文本框（A）”对话框位于“安装地址”节点之上。

STEP 06 在“属性”窗口中，选择 BannerText 属性并键入“安装数据库”。

STEP 07 选择 BodyText 属性并键入“安装程序将在目标机器上安装数据库”。

STEP 08 选择 Edit1Label 属性并键入：服务器名。

STEP 09 选择 Edit1Property 属性并键入 CUSTOMTEXTA1。

STEP 10 选择 Edit1Value 属性并键入：（local）。

STEP 11 选择 Edit2Label 属性并键入：用户名。

STEP 12 选择 Edit2Value 属性并键入：sa。

STEP 13 选择 Edit2Property 属性并键入 CUSTOMTEXTA2。

STEP 14 选择 Edit3Label 属性并键入：密码。

STEP 15 选择 Edit3Property 属性并键入 CUSTOMTEXTA3。

STEP 16 选择 Edit1、Edit2 和 Edit3 的 Visible 属性，并将它们设置为 true；并把 Edit4 的 Visible 属性设置为 false。

15.8 创建自定义操作

STEP 01 在解决方案资源管理器中选择“WebSetupTest”项目。在“视图”菜单上指向“编辑器”，然后选择“自定义操作”。

STEP 02 在自定义操作编辑器中选择“安装”节点，在“操作”菜单上，选择“添加自定义操作”。

STEP 03 在“选择项目中的项”对话框中，双击“Web 应用程序文件夹”。

STEP 04 选择“主输出来自 DBCustomAction（活动）”项，然后单击“确定”按钮关闭对话框。

STEP 05 在“属性”窗口中，选择 CustomActionData 属性并键入“/server=[CUSTOMTEXTA2] /user=[CUSTOMTEXTA3] /pwd=[CUSTOMTEXTA4]”。

15.9 程序部署

STEP 01 生成解决方案。

STEP 02 在“Windows 资源管理器”中，定位到项目目录并找到生成的安装程序。默认路径为\documents and settings\yourloginname\ MyInstaller \project configuration\ WebSetupTest.msi。默认 project configuration 为“Debug”。

STEP 03 将 WebSetupTest.msi 文件和该目录下的其他所有文件和子目录复制到另一台计算机上，或刻录到光盘上制作成安装盘。

STEP 04 双击 Setup.exe 文件就可以运行安装程序（要确认目标计算机上安装了 SQL Server，并有足够权限）。

第 16 章　HLP 文件制作实例入门

本章导读： 帮助文件的制作是软件系统开发不可缺少的一个步骤。HLP 文件是常用的一种帮助文件。下面我们以一个实例来介绍制作 HLP 文件的步骤。

该实例位于光盘的“DEMO\HLP\”目录下。

16.1 RTF 源文件制作

帮助的源文件是 RTF（RichTextFormat）格式的，它存放了所有帮助信息。帮助文件像一本书，由目录及一个个章节组成，各章节之间的跳转是由定义的热点来完成的。我们可以使用 Microsoft Word 来生成 RTF 文件。以下使用的是 Word 2003 版本。

STEP 01 创建文件。

运行 Microsoft Word，选择“文件/新建/空白文档”创建一个新文档，单击“另存为”，并在弹出的对话框中选择“RFT 格式”，命名文件为“Test.RTF”。

STEP 02 设置 Word 显示全部非打印字符。

具体操作为：在“工具”菜单的“视图”选项卡选中“全部”复选框（在“非打印字符”区块）；或者在 Word 2007 中，单击菜单中“Word 选项”按钮，在弹出的“Word 选项”对话框上，选择左侧“显示”项，在右侧的“显示所有格式标记”复选框中打钩。

STEP 03 输入第一节的内容。

目录

软件开发概述

可行性研究

需求分析

系统设计

编码和软件测试

结项和产品化

附录

STEP 04 定义第一节标题。

将光标移到“目录”的前面，选择系统菜单的“插入”，选“脚注和尾注”，选“脚注”和“自定义标记”，输入：$符号，这时文档下方出现脚注编辑窗口，光标也自动落在文档下方的脚注编辑窗口中，在光标处接着输入这一节的标题：“目录”，有时看不到文档下方的脚注窗口，这只要将系统菜单中“视图”的“脚注”选上（Word 2007 中，“插入脚注”命令在“引用”选项卡中）。

STEP 05 定义第一节的标识名。

将光标移到“目录”的前面，选择系统菜单的“插入”，选“脚注和尾注”，选“脚注”和“自定义标记”，输入：#符号，在脚注编辑窗口光标处接着输入这一节的标识名：“Index”。

STEP 06 定义第一节的搜索关健字。

将光标移到“目录”的前面，选择系统菜单的“插入”，选“脚注和尾注”，选“脚注”和“自定义标记”，输入：K 符号，在脚注编辑窗口光标处接着输入这一节的搜索关健字，搜索关健字可以是中文或英文，一个或多个，中间用分号（即“；”）隔开，这里我们输入：“目录；Index”。

STEP 07 定义第一节的浏览顺序编号。

顺序编号相当于一本书中各章节的页码。将光标移到“目录”的前面，选择系统菜单的单的“插入”，选“脚注和尾注”，选“脚注”和“自定义标记”，输入：+符号，在脚注

编辑窗口光标处接着输入这一节的浏览顺序编号，如：0001，0002，0003 等。这里我们输入：0001。

下面把以上用到的符号含义总结一下：

符　号	含　义
#	用于定义该页标识名，页面跳转就是跳到具有指定标识名的页面
$	用于定义该页的标题，这个标题会显示在 Help 的历史窗口中
K	用于定义该页的索引词，各索引词用“；”隔开，用于 Help 的检索
+	用于同类主题的浏览顺序编号

STEP 08 插入第二节。

光标移到文档结尾，选择系统菜单的“插入”，选“分隔符”，“分页符”（在 Word 2007 中，选择“页面布局”选项卡，单击“分隔符\分节符\下一页”）。

STEP 09 添加第二节内容。

在第二页添加“软件开发概述”字样。

STEP 10 第二节脚注标记。

同上面的方法相同，标记第二节脚注。

符　号	标　注
#	Chapter1
$	软件开发概述
K	概述；Summary
+	00002

STEP 11 按照上面的方式完成第 3～8 节内容输入和脚注设置。

STEP 12 完成跳转设置。

- 在第一页中选中“软件开发概述”，选择系统菜单上的“格式/字体”，选择“字体/下画线”，选“双线”。
- 光标在“软件开发概述”后，选择系统菜单上的“格式/字体”，选择“字体/效果”，选“隐藏”，输入跳转目的地的跳转名，这里输入“Chapter1”。设置跳转到第二节。
- 同理设置第一页中的其他跳转。设置完成后如图 16-1 所示。
- 在除第一节外的其他节末尾，添加“返回主题”，并设置跳转到第一节。

STEP 13 添加图像。

在第二节标题下输入：“软件程序员岗位技能图：”，换行后输入“{bmc DEMO.bmp}”。

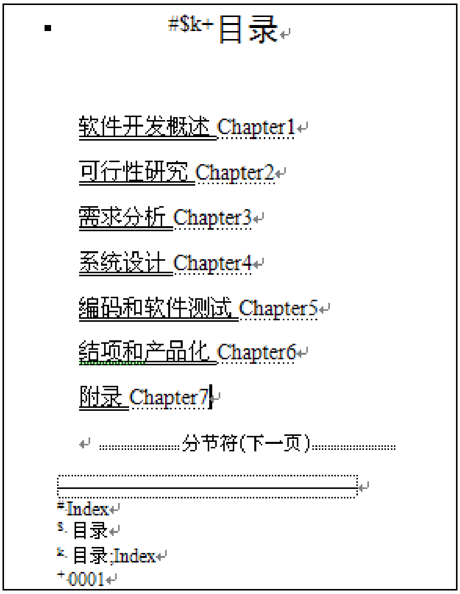


图 16-1　第一节设置

在 RTF 中有两种方法插入图像：直接插入法和语法插入。语法如下：

参　数	功能描述
Bmc	将图像居中显示
Bml	将图像靠左显示
Bmr	将图像靠右显示

注意：图像格式必为 bmp 或 wmf；所有图形文件与 RTF 文件存放同一目录。

STEP 14 定义热点弹出说明。

- 在文档最后新建一节，输入以下文字“图书说明，版权所有”，并设置提跳转标识名称为“about”。
- 选中第一节“目录”，选择系统菜单上的“格式/字体”，选择“字体/下画线”，选“单线”。
- 光标在“目录”后，选择系统菜单上的“格式/字体”，选择“字体/效果”，选“隐藏”，输入跳转目的地的跳转名，这里输入“about”。设置跳转到最后一节。

STEP 15 产生固定的不滚动标题区域。

光标全选不滚动的区域，点取系统菜单上的“格式/段落”，选“正文排列”，将“与下段同页”选上。不滚动区域，在显示特长的信息时很有用。

至此，RTF 文件制作完成！

16.2 编译成为 HLP 文件

接下来我们将把上面产生的 RTF 文件编译为 HLP 文件，使用的工具是 Microsoft Help Workshop，可以在微软网站上下载，最新下载版本是 4.03。

STEP 01 单击 Windows“开始”菜单，选择“程序”→“Microsoft Help Workshop”→“Help Workshop”，单击启动 Help Workshop。

STEP 02 创建工程。

- 单击菜单“File\New”，在弹出的对话框中选择“Help Project”项目，单击“OK”按钮。如图 16-2 所示。
- 在弹出的“Project File Name”对话框上，文件名文本框中输入“Test”，单击保存按钮保存。界面如图 16-3 所示。

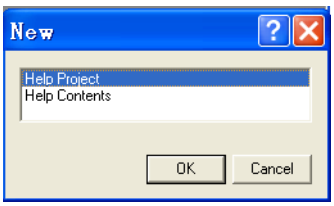


图 16-2 “新建项目”对话框

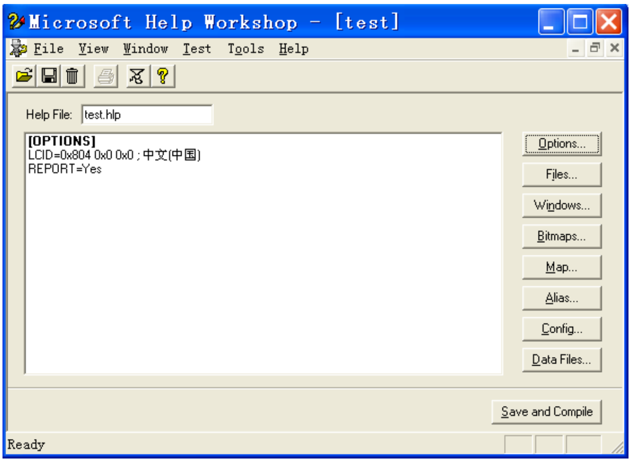


图 16-3 Help Workshop 界面

STEP 03 定制主窗体。

- 单击屏幕右方的“Windows…”按钮，在弹出的对话框上，输入对话框名称“Main”，单击“OK”按钮，如图 16-4 所示。

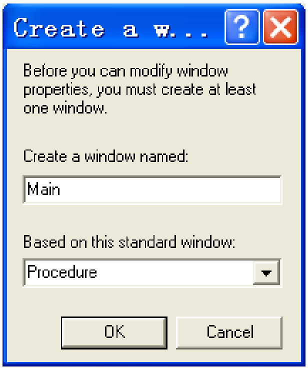


图 16-4 “创建窗体”对话框

- 然后会弹出“Windows Properties”对话框，在“Position”属性页中，单击“Auto-Size”按钮，会出现一个窗口“Help Windows Auto-Size”，如图 16-5 所示。

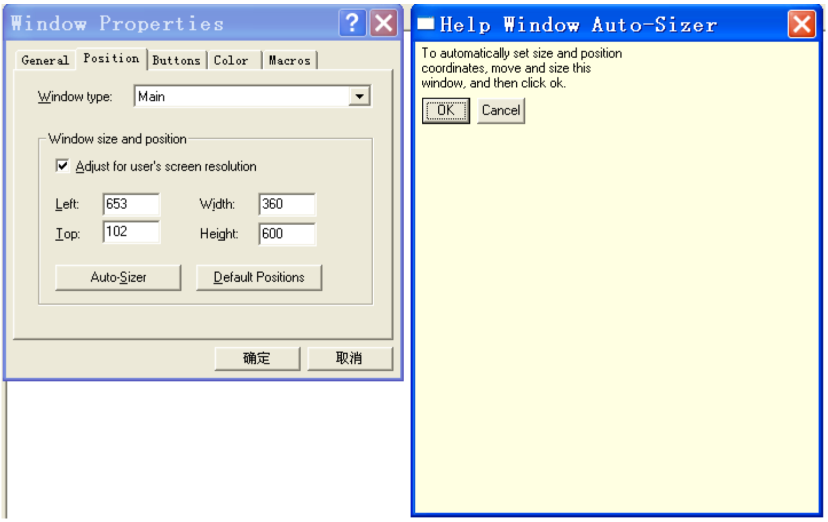


图 16-5 窗体属性设置

- 调整好该窗口大小后，单击该窗口中的“OK”按钮，单击“Windows Properties”窗口中的“确定”按钮返回。可以看到项目文件中添加了以“[Windows]”开头的两行代码。

STEP 04 加入 RTF 文件。

单击屏幕右方的“Option”按钮，选择“Files“选项卡，单击下方的“Change”按钮，会弹出“Topic Files”对话框，单击该对话框上的“Add”按钮，出现“打开”对话框，选择上面编辑好的 RTF 文件“Test.RTF”，如图 16-6 所示。

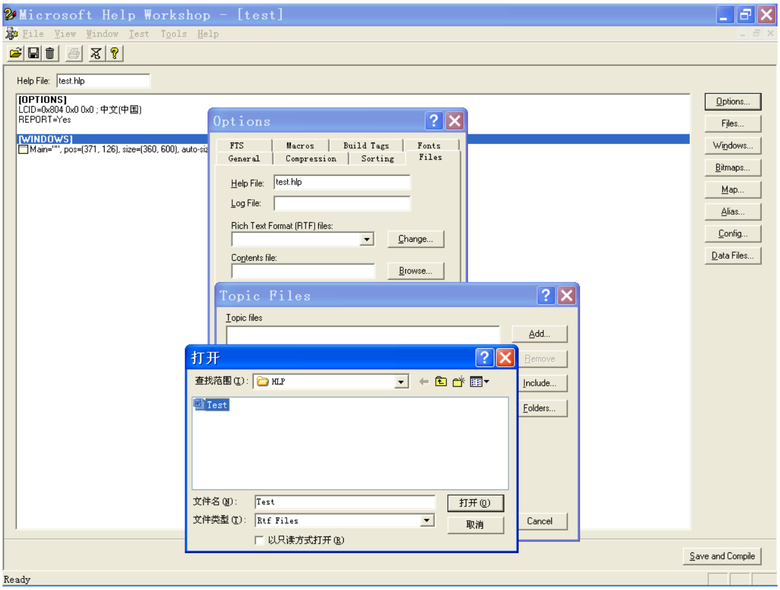


图 16-6 添加 RTF 文件

STEP 05 依次单击“打开”→“OK”→“确定”按钮，返回主窗体。可以看到项目文件中添加了以“[Files]”开头的两行代码，如图 16-7 所示。

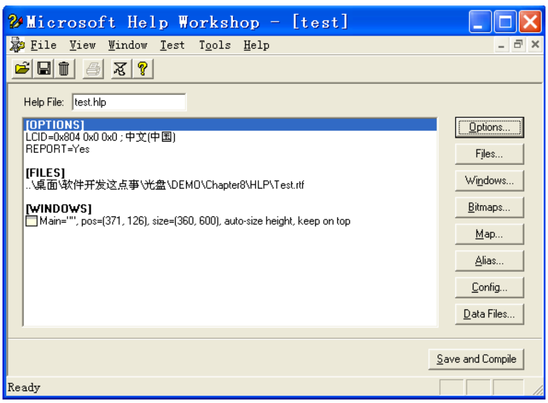


图 16-7 添加 RTF 文件

STEP 06 编译为帮助文件（.hlp）。

单击右侧“Save and Compile”按钮，生成 HLP 文件。可以在对应的目录下找到该文件 Test.HLP。双击可以运行。生成的文件如图 16-8 所示。

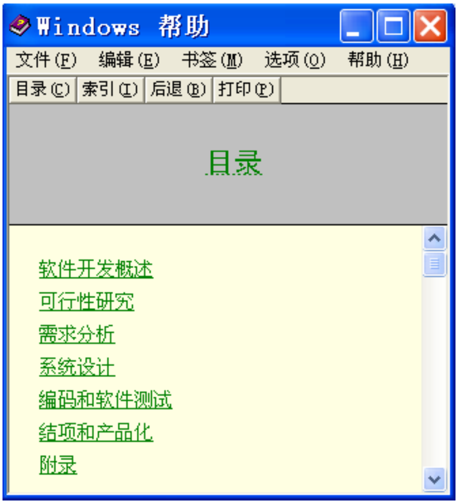


图 16-8　Test.HLP 文件

16.3 制作主题帮助文件（CNT 文件）

STEP 01 启动 Microsoft Help Workshop。单击菜单“File\New”，在弹出的对话框中选择“Help Contents”。

STEP 02 在页面最上方的“Default Filename（and window）”文本框中输入“Test”，在“Default title”文本框中输入“图书说明”。

STEP 03 单击屏幕右侧“Add Above”按钮，在弹出的对话框中，选择“Heading”，在下方“Title”文本框中输入“图书说明”，单击“OK”按钮。

STEP 04 单击“Add Blow”按钮，在弹出的对话框中，选择“Topic”项目，在“Title”中输入“概述”，在“Topic ID”中输入中相对应的“Chapter1”，单击“OK”按钮确认。

STEP 05 同理添加以下 Topic，Title 设定为“可行性研究”、“需求分析”、“系统设计”、“编码和测试”、“结项和产品化”、“附录”，“Topic ID”设定为 Chapter 2 到 Chapter 9。设置后如图 16-9 所示。

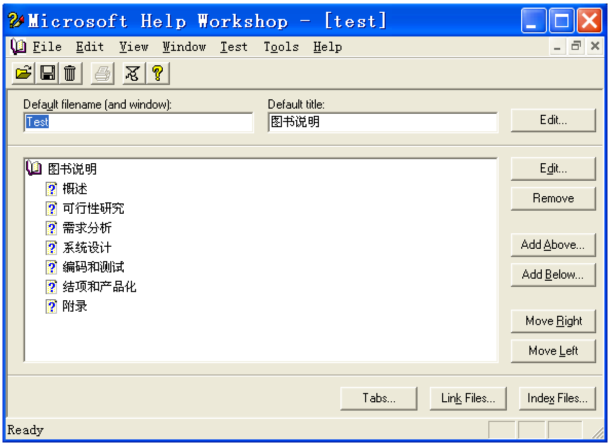


图 16-9　制作 CNT 文件

STEP 06 单击保存按钮，弹出对话框，输入文件名称 Test，保存为“Test.cnt”。

16.4 把 CNT 文件加入工程

STEP 01 打开刚才的工程文件，“Test.hpj”。

STEP 02 单击右侧的“Options…”按钮，选择“Files”属性页，单击“Content File”右侧的“Browse..”按钮，选择刚才创建的 Test.cnt，如图 16-10 所示。

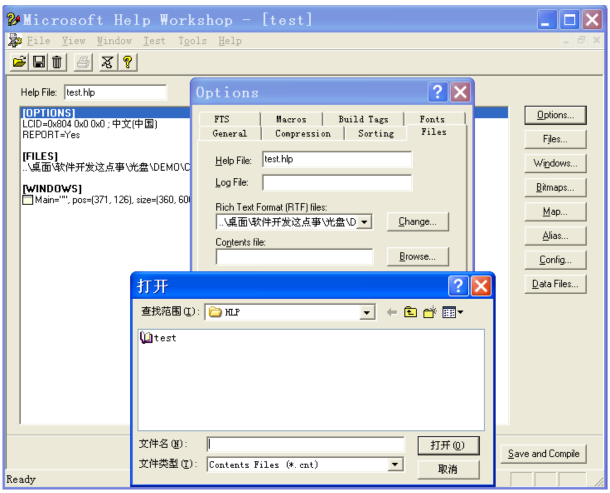


图 16-10　添加 CNT 文件

STEP 03 依次单击“打开”→“确定”按钮返回主窗体。可以看到项目文件中“[Options]”节最后添加了以“CNT=...”开头的一行代码。

STEP 04 单击右侧“Save and Compile”按钮，重新生成 HLP 文件。

STEP 05 运行新生成的 HLP 文件。出现新的主题帮主文件，如图 16-11 所示。

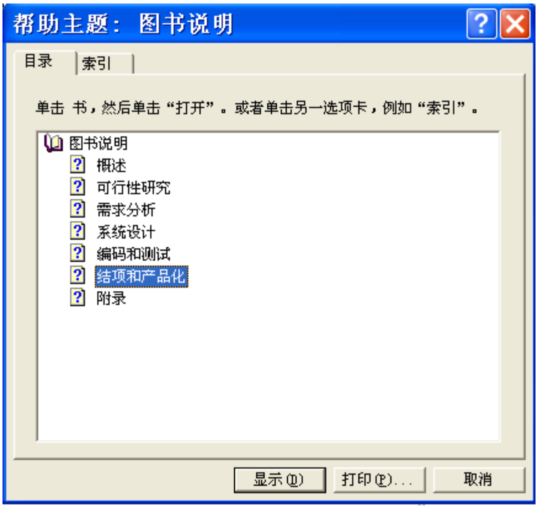


图 16-11　生成的 HLP 文件

第 17 章　CHM 文件制作实例入门

相对于 HLP 文件来说，CHM 文件格式更加丰富。在 CHM 文件的属性中我们可以看到 CHM 的类型是“已编译的 HTML 帮助文件”，其实 Internet Explorer 支持的 JavaScript、VBScript、ActiveX、Java Applet、Flash、HTML 图像文件（GIF、JPEG、PNG）、音频视频文件（AU、MIDI、WAV、AVI）等，CHM 文件也同样支持。正因为 CHM 支持如此多的特性，理所当然地成为了制作软件帮助文件乃至电子书的首选格式。

制作 CHM 文件的最佳工具是 Microsoft HTML Help Workshop。可以在微软的网站上下载该软件，软件版本是 1.32，也可以从网络上下载汉化补丁。

该实例位于光盘的“DEMO\CHM\”目录下。

17.1 材料准备

开始制作 CHM 文档之前，我们必须准备好“原材料”，即 HTML 格式的网页文件。制作 HTML 文件的软件很多，我们最常用的有 FrontPage、Dreamweaver 等。这些原材料可以在本书配套光盘“DEMO\CHM\HTML Files”下找到。

17.2 创建工程

STEP 01 单击 Windows“开始”菜单中，选择“程序→HTML Help Workshop”程序组中的“HTML Help Workshop”快捷方式。单击启动应用程序。

STEP 02 单击“File\New”，弹出“New”对话框，选择“Project”，如图 17-1 所示。

STEP 03 单击“OK”按钮，弹出“New Project”对话框，询问是否从 WinHelp 项目转换，如图 17-2 所示。

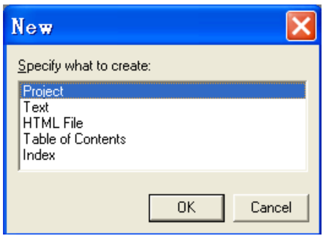


图 17-1 “新建项目”对话框

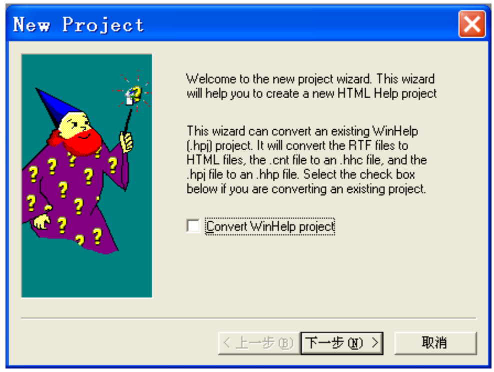


图 17-2 “WinHelp 转换”对话框

STEP 04 单击“下一步”按钮，弹出“New Project--Destination”对话框，设置项目文件的保存位置。单击“Browse…”按钮，在弹出的“打开”对话框中，选择项目存放位置，在“文件名”文本框中输入“Test”，单击“打开”按钮。可以看到“Browse…”按钮上的文本框中生成了要保存的路径文件名，如图 17-3 所示。



图 17-3 “设置保存路径”对话框

STEP 05 单击“下一步”按钮，弹出“New Projec -- Existing Files”对话框，如果已经制作了 hhc、hhk 和 htm 文件，可以勾选上面的相应项。因为我们没有制作目录文件和索引文件，就只选中“HTML files”，如图 17-4 所示。

STEP 06 单击“下一步”按钮，弹出“New Project--HTML Files”对话框，单击“Add”按钮，弹出“打开”对话框，选中所有要添加的 HTML 文件，单击“打开”按钮。把所有制作好的 HTML 文件都添加进去，如图 17-5 所示。



图 17-4 “添加现有文件”对话框

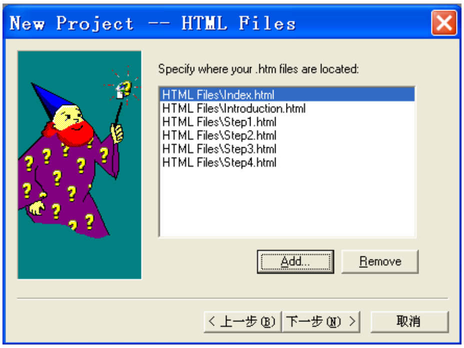


图 17-5 添加 HTML 文件

STEP 07 单击“下一步”按钮，弹出“New Project--Finish”对话框，如图 17-6 所示。



图 17-6 “完成”对话框

STEP 08 单击“Finish”按钮完成向导，进入主界面，如图 17-7 所示。可以到有“Project”、“Contents”、“Index”三个属性页，分别对应“工程”、“内容页”、“索引页”。

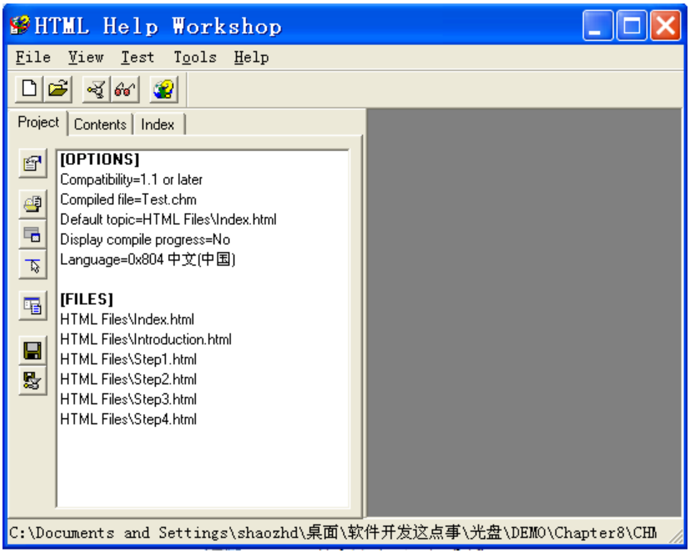


图 17-7　HTML Help Workshop 主界面

17.3 设置 CHM 文档的标题

STEP 01 选取“Project”属性页，单击“Add/Modify window definitions”按钮（左边从上数第三个），弹出“Add a New Window Type”，输入“Main”，如图 17-8 所示。

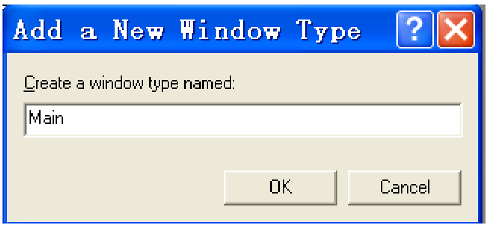


图 17-8　“添加新窗体”对话框

STEP 02 单击“OK”按钮。弹出“Window Types”对话框，在“Title bar text”文本框中输入“HLP 文件制作实例入门”，如图 17-9 所示。

STEP 03 在“Window Type”对话框中，切换到“Position”页面，点击“AutoSizer”按钮，弹出“Autosizer”窗口，该窗口为最终 CHM 文件窗口大小，如图 17-10 所示。

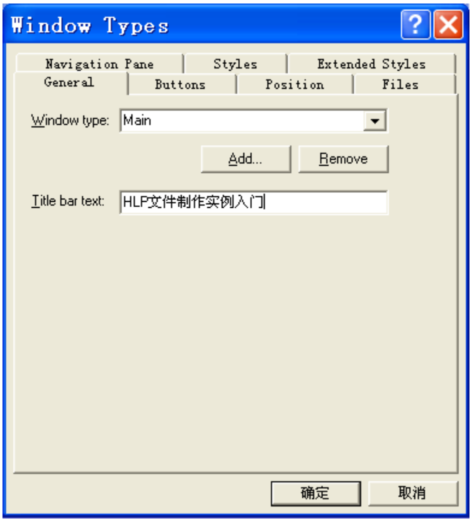


图 17-9 “窗体属性设置”对话框

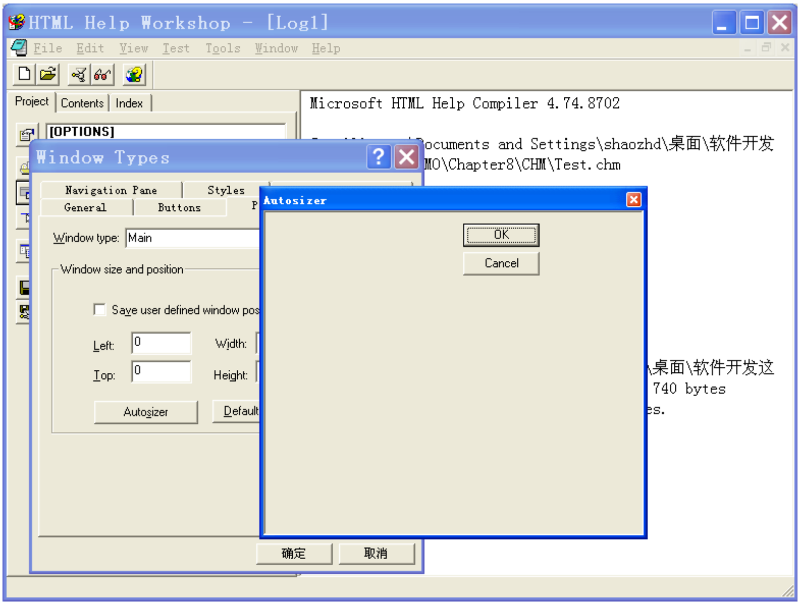


图 17-10 窗体大小设置

STEP 04 调整该窗口大小到满意为止，单击窗口上“OK”按钮返回。

STEP 05 选择“Window Types”对话框上的“Files”属性页，确认“Default”文本框中为“HTML Files\Index.html”，如图 17-11 所示。

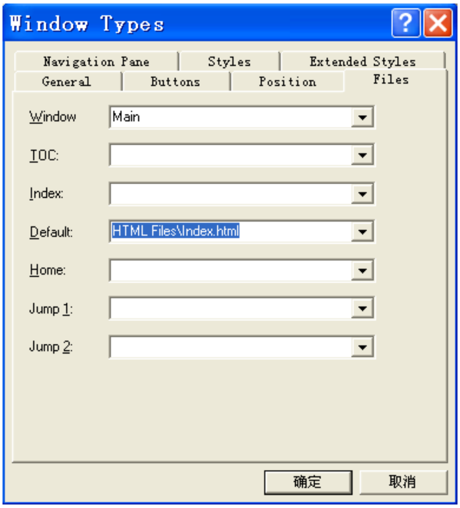


图 17-11　设置窗体对应 HTML 文件

STEP 06 单击“确定”按钮返回主窗口。单击屏幕左侧最下方的“save all files and compile”按钮，启动编译。可以看到在项目目录下生成了“Test.chm”文件。双击可以运行该文件，界面如图 17-12 所示。

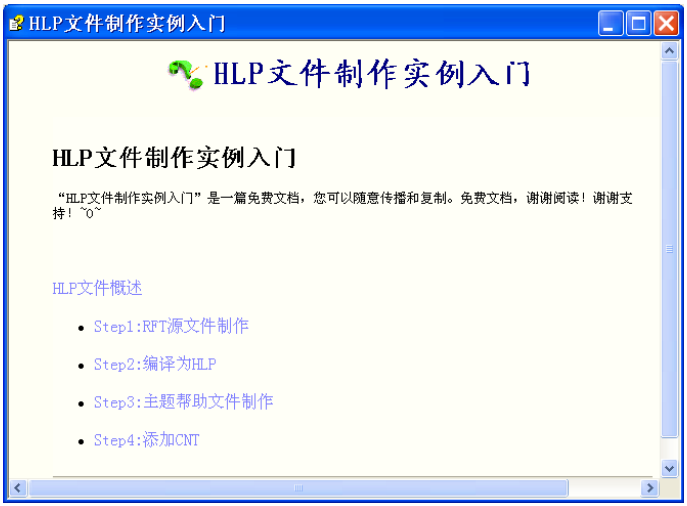


图 17-12　生成的 CHM 文件

可以看到 CHM 文件中包含了网页的所有信息，就如同将网站打包一样。但是可惜没有导航窗口和功能按钮。

17.4 修改文档对话框属性

STEP 01 选取“Project”属性页，点击“Add/Modify window definitions”按钮（左边从上数第三个），选取“Navigation Pane”属性页。选中“Windows with navigation pane…”前边的复选框，并选中“Auto sync”和“Search tab”、“Favorites tab”，如图 17-13 所示。

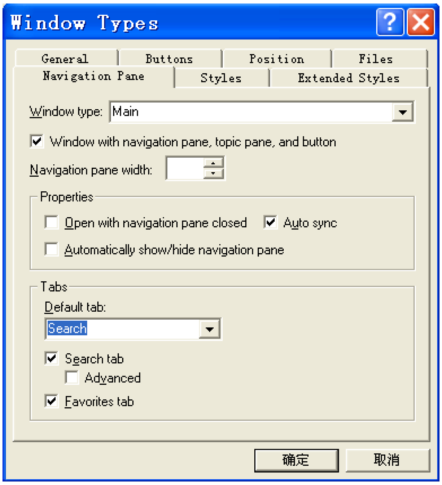


图 17-13 窗体导航属性设置

STEP 02 单击“确定”按钮返回主界面。单击屏幕左侧最下方的“save all files and compile”按钮编译。打开新生成的 Test.chm，发现界面上多了“搜索”和“书签”面板，如图 17-14 所示。

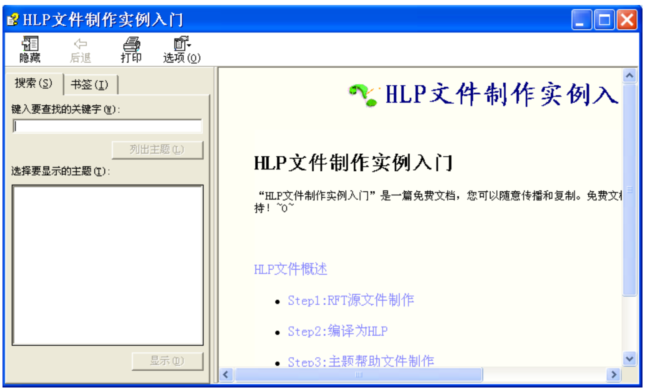


图 17-14 带“搜索”和“书签”的 CHM 文件

17.5 制作目录

STEP 01 单击主窗体中“Contents”选项卡，会弹出一个对话框，选择是否创建目录文件，如图 17-15 所示。



图 17-15 询问是否创建目录

STEP 02 选择“Create a new contents file”，单击“OK”按钮。弹出“另存为”对话框，在“文件名”文本框中输入“Test”，单击“OK”按钮。可以主程序切换到“Contents”页面，如图 17-16 所示。

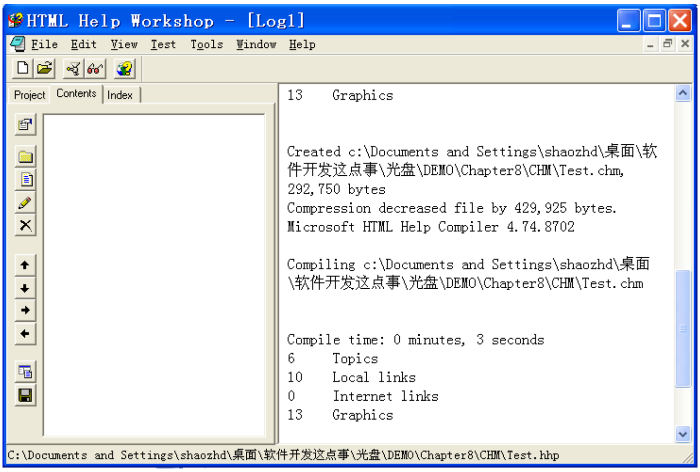


图 17-16 “Contents”页面

STEP 03 单击“Insert a page”按钮（左边从上数第三个），弹出“Table of Contents Entry”对话框。在该对话框上“Entry Title:”文本框中输入“HLP 文件制作实例入门”；单击“Add”按钮，弹出“Path or URL”对话框，选择“Index.html”如图 17-17 所示。

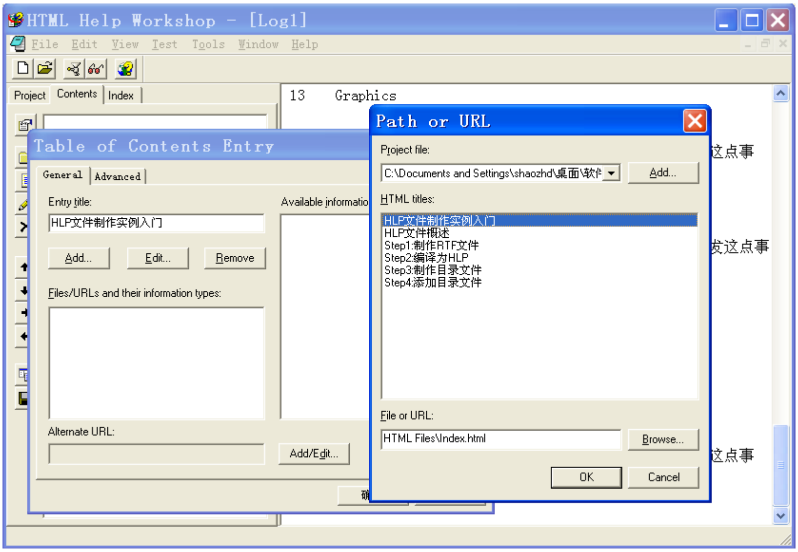


图 17-17 “文件选择”对话框

STEP 04 依次单击“OK”、“确定”按钮，返回主窗体；可以看到刚才添加的项显示在窗体中。

STEP 05 同理添加其他项。添加后界面如图 17-18 所示。

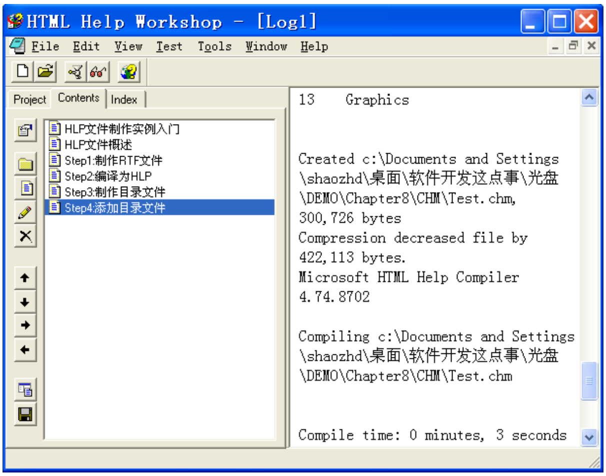


图 17-18 添加目录后

STEP 06 选中第 2 个项目，单击“Move Selection Right 按钮”（左边从上数第 8 个）。

STEP 07 选中第一项，单击“Edit selection”按钮，在弹出的“Table of Contents Entry”

对话框上，选择“Advanced”属性页，在“Image Index:”后输入“1”，如图 17-19 所示。

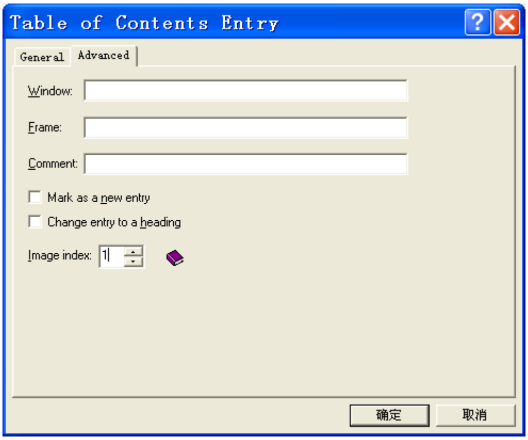


图 17-19　设置目录图像标识

STEP 08 单击“确定”按钮返回。

17.6 添加目录

STEP 01 选取“Project”属性页，单击“Add/Modify window definitions”按钮，选取“Files”属性页，在“TOC:”文本框中输入“Test.hhc”，如图 17-20 所示。

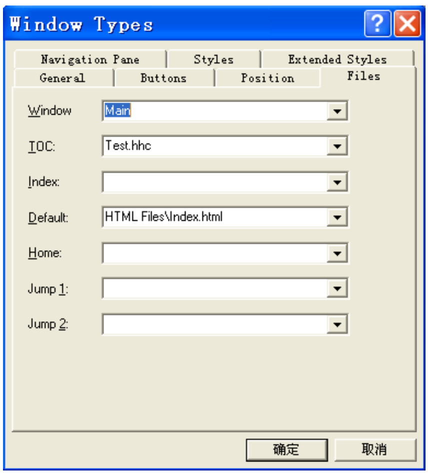


图 17-20　设置“TOC”属性

STEP 02 切换到“Navigation Pane”属性页，在“Default tab:”下的下拉框中选择“Contents”，如图 17-21 所示。

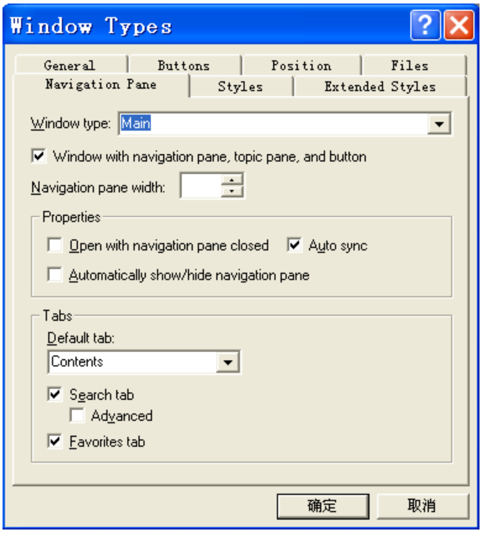


图 17-21　修改“Navigation Pane”属性

STEP 03 单击“确定”按钮返回，重新编译，打开新产生的“Test.chm”，如图 17-22 所示可以看到目录添加成功！

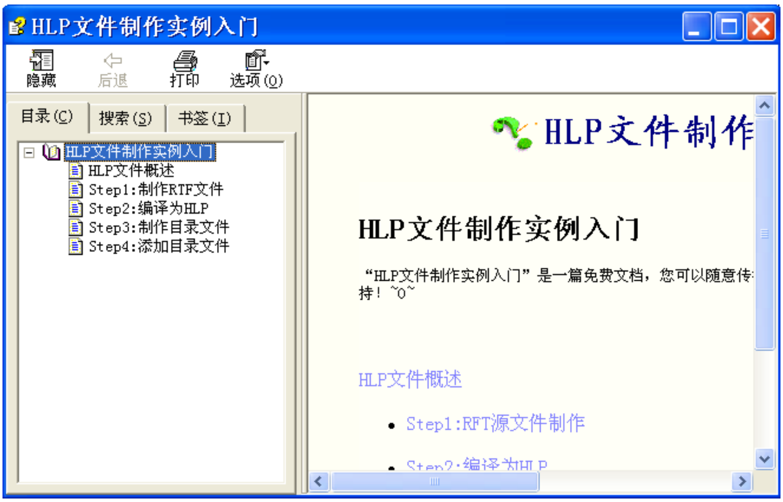


图 17-22　添加目录后的 CHM 文件

17.7 制作索引

STEP 01 单击主窗体中“Index”选项卡，会弹出一个对话框，选择是否创建索引文件，如图 17-23 所示。



图 17-23　询问是否创建索引

STEP 02 选择“Create a new index file”，单击“OK”按钮。弹出“另存为”对话框，在“文件名”文本框中输入“Test”，单击“OK”按钮。可以主程序切换到“Index”页面。

STEP 03 单击“Insert a keyword”（左边第二个按钮），弹出“Index Entry”对话框；在“keyword”文本框中输入“概述”，单击“Add”按钮，选择“Introduction.html”，如图 17-24 所示。

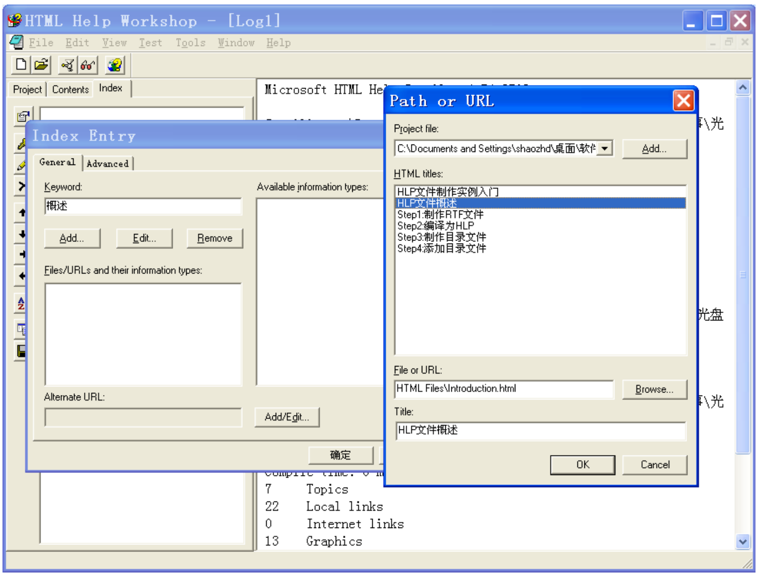


图 17-24　选择文件

STEP 04 依次单击“OK”、“确定”按钮返回主窗体，可以看到已经添加了一个关键词。

STEP 05 同理添加其他索引，如图 17-25 所示。

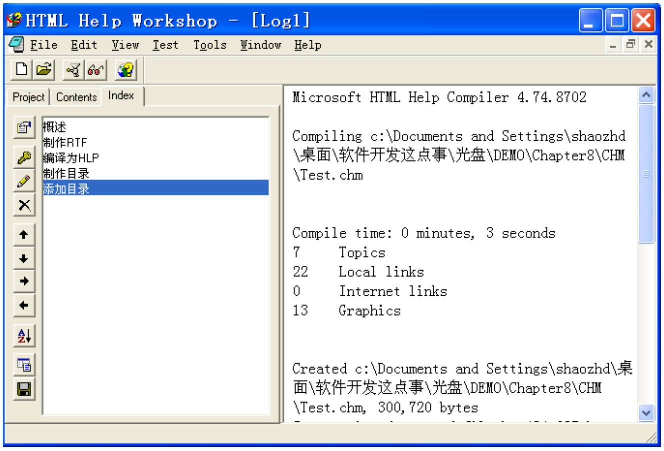


图 17-25　添加索引后

17.8 添加索引

STEP 01 取“Project”属性页，单击“Add/Modify window definitions”按钮，选取“Files”属性页，在“Index:”文本框中输入“Test.hhk”。

STEP 02 重新编译，打开新生成的文件，如图 17-26 所示。索引制作成功！

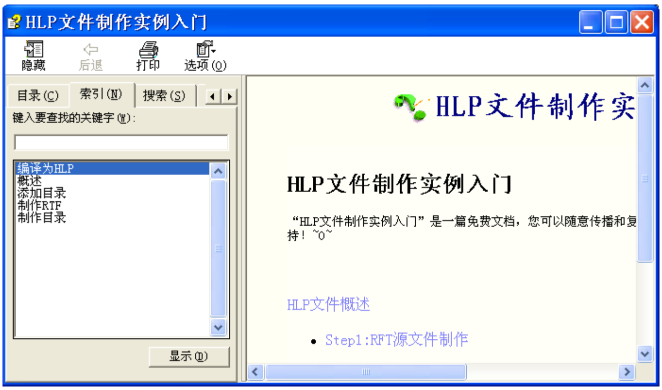


图 17-26　添加索引后的 CHM 文件

17.9 说明

在制作过程中用到了.hhp、.hhk 和.hhc 三种文件，它们的作用分别如下：

- .hhp：保存项目文件配置的有关内容，方便以后生成 chm 的语言以及打开 chm 的起始页。
- .hhk：保存关键词索引目录的内容，其实是一个 html 文件，可以用记事本等打开查看和编辑。
- .hhc：保存内容页的配置。其实是一个 html 文件，可以用记事本等打开查看和编辑。

第18章　Visual SourceSafe实例入门

Microsoft Visual SourceSafe 是最常用的源代码控制工具，它能够和 Visual Studio 完美集成，是团队开发必不可少的工具之一。

18.1　软件安装

1. 可以在 Visual Studio 6.0 的安装盘中找到 Visual SourceSafe，也可以单独购买该软件。

2. 程序的安装很简单，可以使用默认设置，一路单击“下一步”就可以完成。安装完成后可以在 Windows“开始”菜单“程序”菜单中找到“Microsoft Visual SourceSafe”程序组。

18.2　服务端创建 VSS 数据库和用户

STEP 01 启动 VSS 管理器。单击“Microsoft Visual SourceSafe”程序组中的“Visual SourceSafe 6.0 Admin”菜单项，出现如图 18-1 所示的界面。

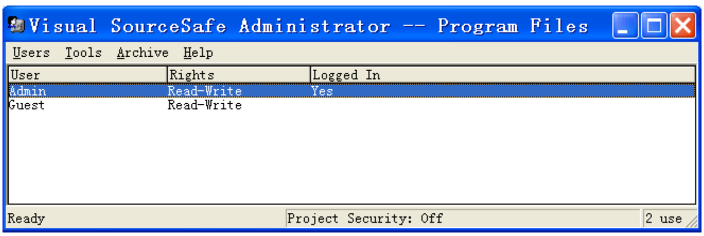


图 18-1　Visual SourceSafe 6.0 Admin 界面

STEP 02 创建 VSS 数据库。单击下拉菜单“Tools”中的“Create Database...”菜单项，将出现如图 18-2 所示的窗口，选择新数据库的路径，输入数据库名“MyData”，然后单击 OK 按钮，系统将开始创建工作。完成后会弹出一个对话框提示“Database Creation Complete”，单击“OK”按钮即可。

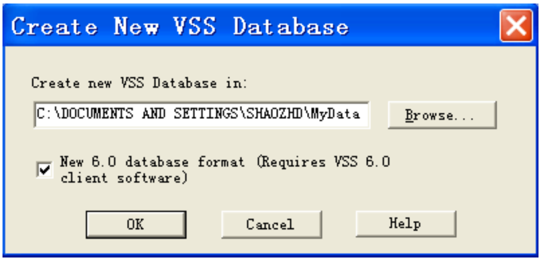


图 18-2　创建新数据库

完成后，打开该文件夹，可以看到对应生成了“MyData”文件夹，同时该文件下包含两个文件和三个目录。

名　　称	说　　明
srcsafe.ini	SourceSafe 数据库配置文件
users.txt	用户信息
data	存放具体数据库
users	存放用户具体配置信息
temp	临时目录

文件夹如图 18-3 所示。

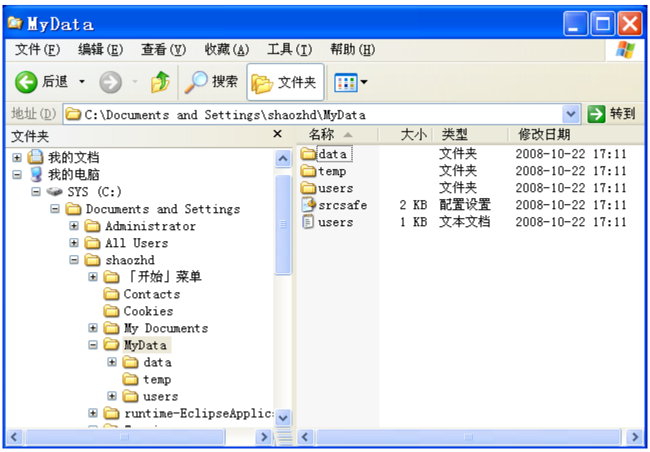


图 18-3　SourceSafe 文件格式

STEP 03 选中新创建的数据库。单击下拉菜单"Users/Open SourceSafe Database..."，弹出"Open SourceSafe Database"对话框，单击"Browse…"按钮，选择上面创建的数据库 MyData。依次单击"OK"、"Open"打开该数据库，如图 18-4 所示。

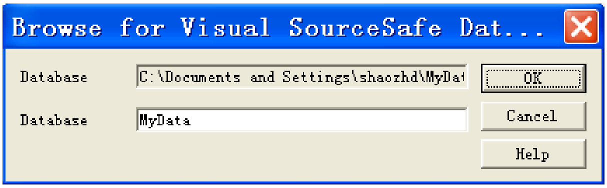


图 18-4　选择 SourceSafe 数据库

系统会弹出如图 18-5 所示的对话框，提示当前用户 Admin 没有密码，为安全起见，建议修改管理员密码，单击"OK"按钮。

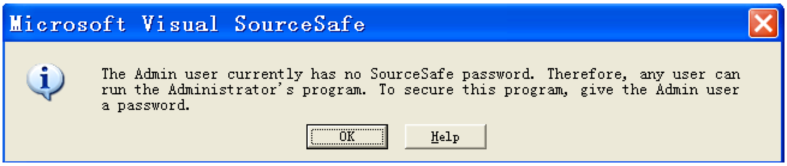


图 18-5　空密码提示

STEP 04 创建用户。选择菜单项"Add User..."，输入用户名和口令，创建新用户，如图 18-6 所示。同样的方法为项目组内每个成员创建一个账户。

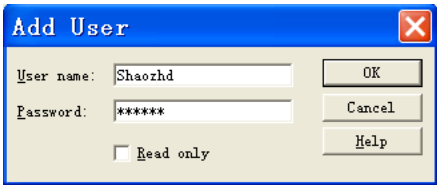


图 18-6　"添加用户"对话框

当 VSS 服务器端的安装配置工作全部完成，并在客户端也完成了 VSS 的安装，这样就可启用 VSS 来管理控制整个开发小组的源代码和文档的版本管理。

18.3 客户端选择数据库

STEP 01 单击系统 SourceSafe 安装项中"Microsoft Visual SourceSafe 6.0"菜单项，

启动 SourceSafe，如图 18-7 所示。



图 18-7　Microsoft Visual SourceSafe 6.0 界面

STEP 02 单击下拉菜单 File 中“Open SourceSafe Database...”，弹出“Open SourceSafe Database”对话框，单击“Browse…”按钮，选择刚才创建的 MyApp 数据库（可以把服务器上的 MyApp 文件夹共享，在这个地方浏览网络路径，打开此数据库），如图 18-8 所示。

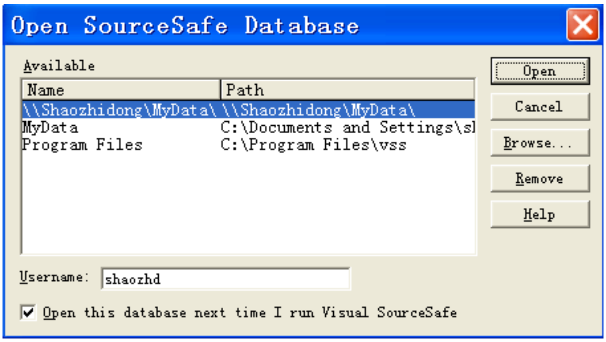


图 18-8 “选择数据库”对话框

18.4 添加项目到 VSS 数据库

以下操作前，首先要在 SourceSafe 中完成选择数据库的操作。

18.4.1　通过 SourceSafe 添加项目

STEP 01 运行 SourceSafe，单击菜单 File 中的命令“Create Project...”，在弹出的对话框上“Project”文本框中输入“TestProject”，“Commment：”文本框中输入“这是一个测试项目。”。单击“OK”按钮创建该项目，如图 18-9 所示。可以看到在左上方栏目中已经创建了该项目，如图 18-10 所示。同理可以创建多个项目。

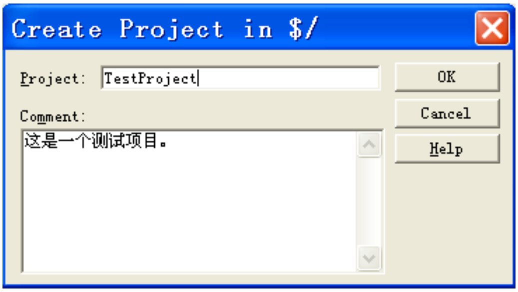


图 18-9　“添加新项目”对话框



图 18-10　创建新项目后的主界面

STEP 02 添加文件。选中“TestProject”项目，单击“File”中“Add Files”命令，将跳出对话框，选择相应文件或目录，单击 Add 按钮，将它们添加到 TestProject 中去，如图 18-11 所示。

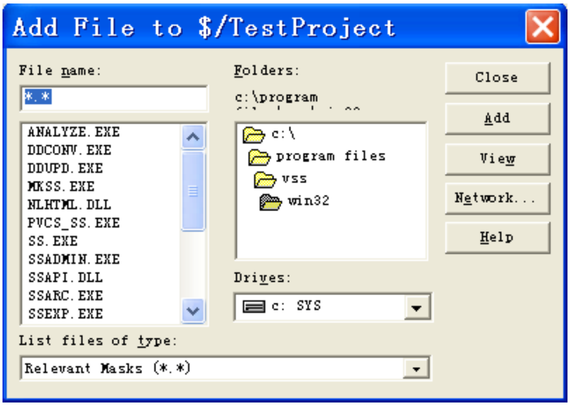


图 18-11 “添加文件”对话框

18.4.2 通过 VisualStudio.Net2005 集成开发环境添加项目

STEP 01 启动 VisualStudio.net2005，并打开或新创建一个工程。

STEP 02 安装 SourceSafe 后，VS.NET 开发环境会自动在“文件”菜单下添加“源代码管理”菜单项。如果没有的话，可以单击“工具”菜单上的“选项……”，弹出“选项”对话框。在对话框上左侧列表中，选择“源代码管理/插件选择”，选择“Microsoft Visual SourceSafe”如图 18-12 所示。

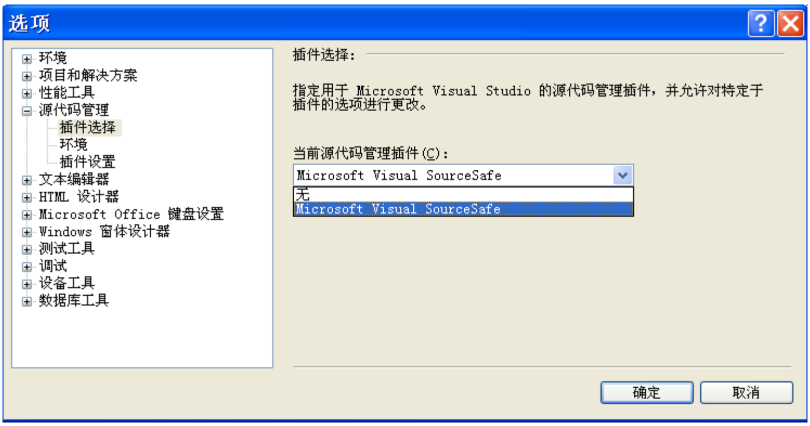


图 18-12 启动 SourceSafe 控制

STEP 03 在 VS.NET“解决方案管理器”中选中当前解决方案，右键单击该解决方案，在弹出的菜单中选择“将解决方案添加到源代码管理……”。

STEP 04 在弹出的“Add to SourceSafe Project”对话框中，输入工程名，并选择存放在根节点下，如图 18-13 所示。

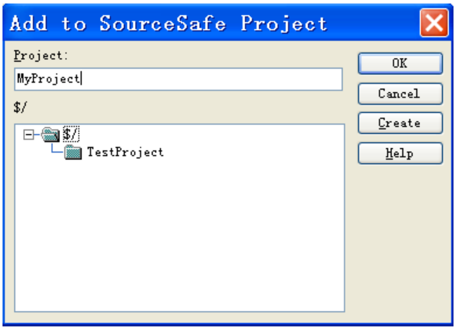


图 18-13 把项目加入 SourceSafe

STEP 05 单击“OK”按钮，会提示该项目不存在，问是否要创建，单击“Yes”按钮，如图 18-14 所示。

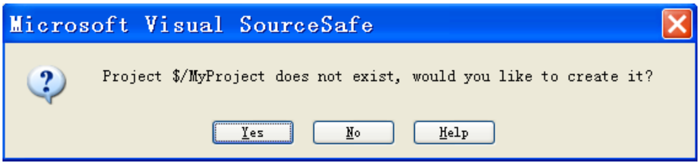


图 18-14 询问是否创建

STEP 06 从系统“开始”菜单“程序”中运行“Microsoft SourceSafe 6.0”，可以看到当前解决方案中的所有文件都被添加到 SourceSafe 中，如图 18-15 所示。

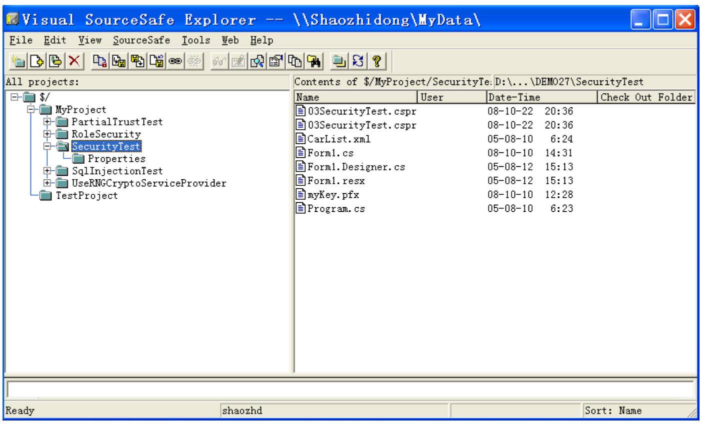


图 18-15 加入项目后的 SourceSafe 主界面

在把解决方案添加到 SourceSafe 数据库中后，其他项目组的人员就可以使用该项目了。

18.5 在客户端 VS.NET 开发环境中使用 VSS

STEP 01 打开 SourceSafe 数据库中的项目。运行 VisualStudio.NET2005，选择“文件”菜单下的“源代码管理\在源代码管理中打开……”，在弹出的“Create Local project from SourceSafe”对话框上，输入要保存项目的本地路径，并选择下方“MyProject”。单击“OK”按钮，如图 18-16 所示。

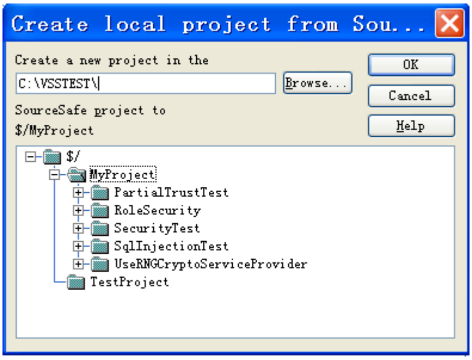


图 18-16 打开 SourceSafe 数据库中的项目

STEP 02 可以看到，解决方案中所有的文件都复制到了本地刚才指定的文件夹中，并且在 VS.NET 开发环境“解决方案管理器”中可以看到这些文件签名都有一把小锁，表明文件处于未打开状态，如图 18-17 所示。

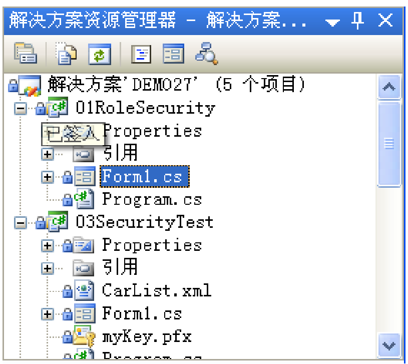


图 18-17 文件保护状态

STEP 03 右键单击解决方案，选择“签出以进行编辑……”项。在弹出的对话框中选择要修改的文件，并输入相应注释（当我们对解决方案中的文件进行修改时，VS.NET 自动把该文件从数据库中读出（Check Out）），如图 18-18 所示。

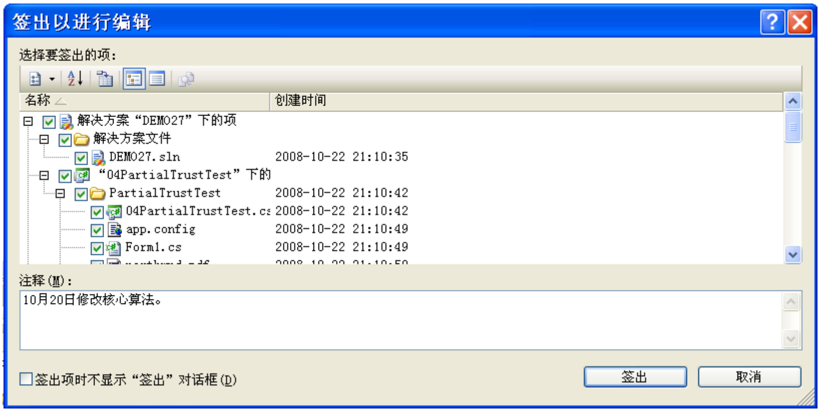


图 18-18 “文件签出”对话框

STEP 04 单击“签出”，VS.NET 自动把文件从 VSS 数据库中读出，这时可以看到“解决方案管理器”中的签出的文件前有了一个红色的对钩标记，如图 18-19 所示。

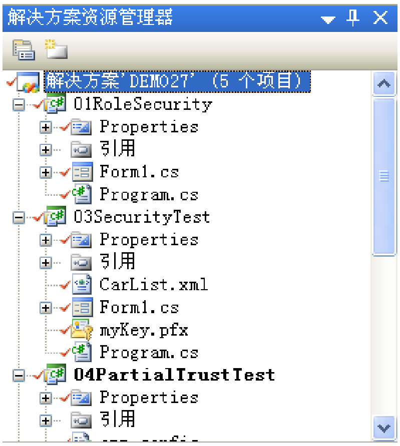


图 18-19 签出后的文件状态

STEP 05 此时再从“开始”菜单中启动 SourceSafe，可以看到正在修改的文件被红色标记，并能看到使用人的用户名，签出（Check out）的目录等信息，如图 18-20 所示。

默认情况下，对于每个文件，在同一时刻，SourceSafe 只允许一个人作签出修改（Check Out）。

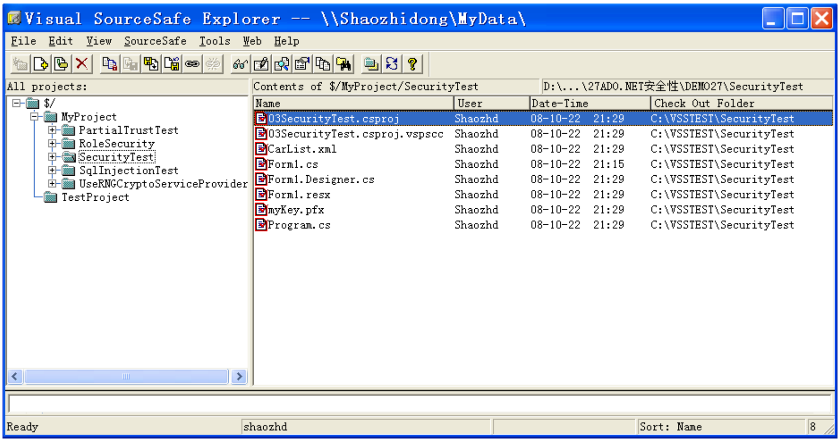


图 18-20　文件签出后 SourceSafe 中的显示

STEP 06 签入数据。在 VS.NET 开发环境中修改完指定代码后，可以右键单击解决方案文件或者指定文件，选择“签入……”菜单项，把所做的修改更新回（Check in）VSS 数据库。

18.6 修改历史查看

STEP 01 运行 SourceSafe，右键单击指定文件，选择“Show History…”菜单项，如图 18-21 所示。

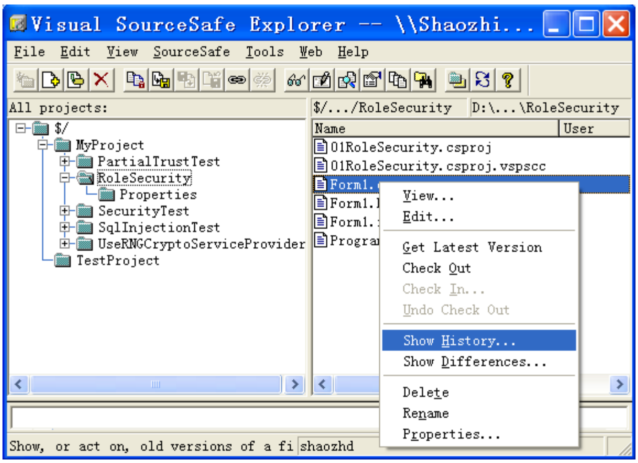


图 18-21　查看历史

STEP 02 在弹出的“History Option”对话框上，单击“OK”按钮，如图 18-22 所示。

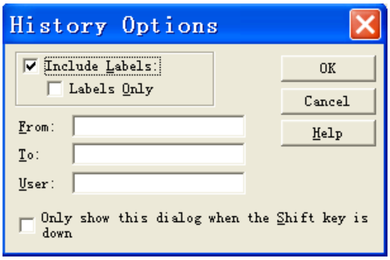


图 18-22　历史选项

STEP 03 在弹出的修改历史对话框中单击“Details…”可以查看修改时添加的注释，如图 18-23 所示。

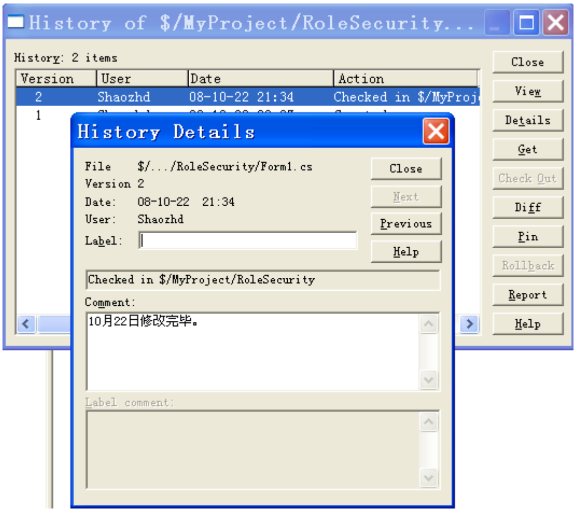


图 18-23　文件修改历史

18.7 用户权限修改

STEP 01 从 Windows“开始”菜单中运行“Visual SourceSafe 6.0 Admin”。

STEP 02 单击“Tools\Options…”菜单项，弹出“SourceSafe Options”对话框，切换到“Project Security”属性页面，在“Enable Project Security”复选框前打钩，如图 18-24 所示。

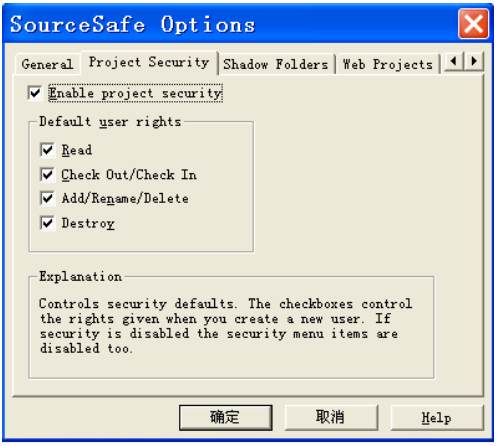


图 18-24 “用户权限设置”对话框

可以看到 SourceSafe 共有 4 种权限：

- Read：只读。
- Check Out/Check In：可以签出、签入。
- Add/Rename/Delete：可以添加、重命名、删除。
- Destory：可以破坏文件。

STEP 03 选择“Tools”下“Rights by Project”菜单项，出现“Project Rights”对话框。我们可以在此为各个用户设置访问指定项目或文件的权限（如果一个解决方案中，有多个项目，每个项目组成员负责不同的项目，则此设置非常有用，我们可以设置为每个人对自己负责的项目有全部权限，而对别人负责的项目只有只读权限），如图 18-25 所示。

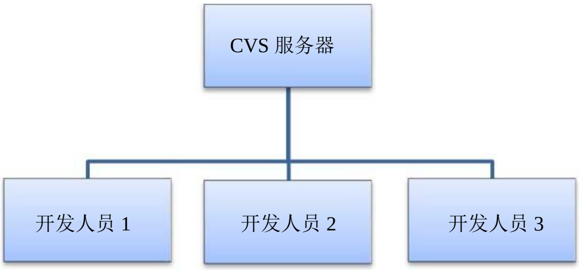


图 18-25 项目权限修改

第 19 章　CVS 实例入门

在源代码版本管理工具当中，CVS 的使用非常普遍。CVS 是一个 C/S 系统，项目组内的开发人员可以通过 CVS 服务器来记录文件版本，从而达到保证文件同步的目的。

CVS 的工作模式如下：



Eclipse 内建了 CVS 客户端，方便团队 CVS 控制；目前 UNIX 平台都带 CVS 服务器，Windows 平台比较流行的是 CVSNT。由于有很多的书籍和文章介绍 UNIX 和 Linux 平台上 CVS 的使用，因此本文不再做相关介绍，而着重介绍 Windows 平台上 CVS 的使用。

接下来我们以一个简单的实例介绍在 Windows 环境中 CVS 的使用。

19.1 安装 CVSNT

STEP 01 可以从 http://www.march-hare.com/cvsnt/下载最新版本的 CVSNT。本文使用的软件版本是 CVSNT2.5.03.2382。

STEP 02 双击下载文件“cvsnt-2.5.03.2382.msi”，运行安装文件，安装 CVSNT。

安装后按照程序提示重新启动计算机。

STEP 03 在 Windows“开始”菜单“程序/CVSNT”中运行“CVSNT Control Panel”项。启动“CVS”对话框。选择“About”属性页，可以看到 CVS 服务正在运行，如图 19-1 所示。

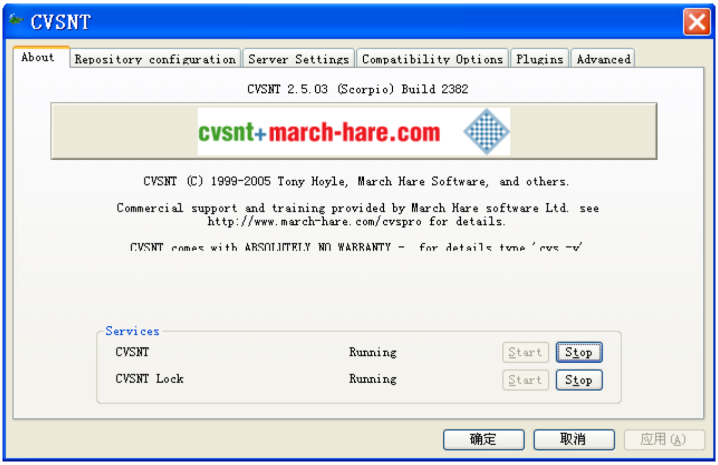


图 19-1　CVSNT 界面

STEP 04 为了使 Eclipse 能够与 CVSNT 正常通信，还要将“Compatibility Options”属性页中的“Respond as cvs 1.11.2 to version request”项选中，如图 19-2 所示。

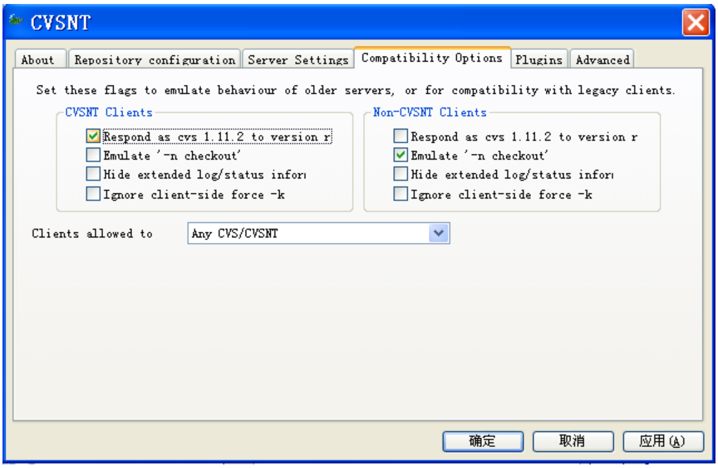


图 19-2　CVSNT 兼容性设置

19.2 服务端创建仓库和用户

STEP 01 在以上 CVSNT 对话框中，选择“Repositories Configuration”属性页，单击“Add”按钮创建新的 CVS Repository。在弹出的“Server Settings”对话框上，在“Location:”文本框中输入文件目录“C:/CVSTest”，在下面的“Name:”文本框中输入“/CVSTest”（CVS 中的 Repository 相当于 SourceSafe 中的 Database），如图 19-3 所示。

STEP 02 单击“OK”按钮，弹出“cvsnt”对话框，询问是否要初始化目录，如图 19-4 所示。

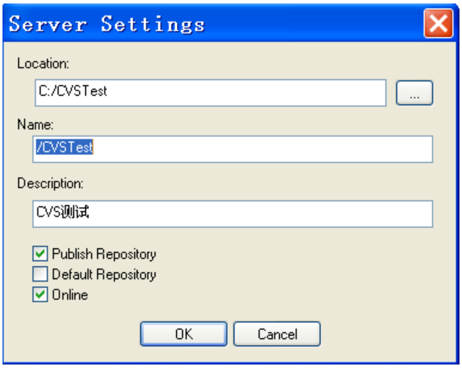


图 19-3　创建 CVS 仓库

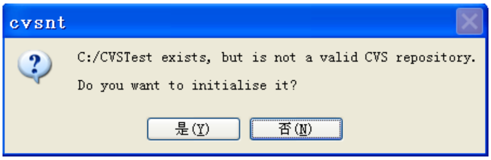


图 19-4　是否初始化目录

单击“是”返回。这样就创建完成指定的 Repository。

STEP 03 创建用户：打开系统“控制面板”下的“用户账号”项，创建一个新的账户“CVSUser1”，并指定其密码。（CVSNT 最简单的用户管理方式就是和 NT 验证结合，这样，创建一个 NT 用户也就创建了一个 CVS 用户。可以通过设置该用户的访问权限来控制用户对 CVS 的访问）

19.3 添加项目到 CVS 服务器

STEP 01 运行 Eclipse。创建一个名为“TestProject”的 Java 项目，在“src”目录中添加 Main.java。如图 19-5 所示。

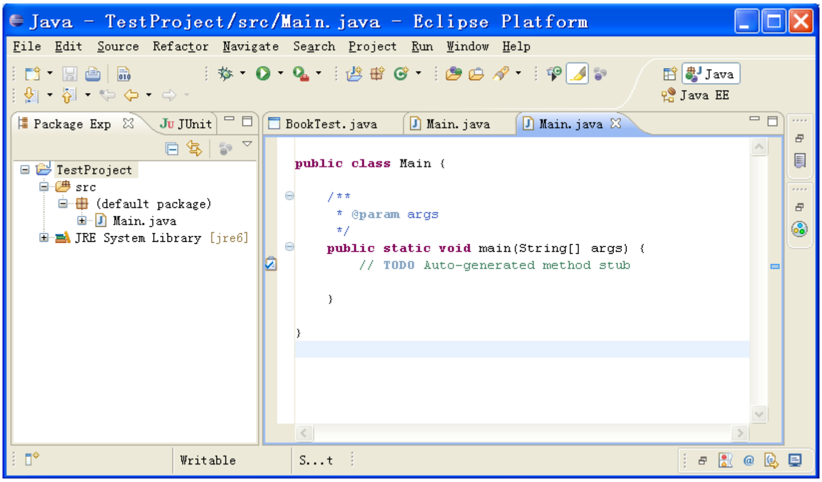


图 19-5　Eclipse 主界面

STEP 02 在 Eclipse 主窗体中，单击“Window”菜单下的“Open Perspective/Other…”，在弹出的“Open Perspective”对话框中选择“CVS Repository Exploring”，如图 19-6 所示。

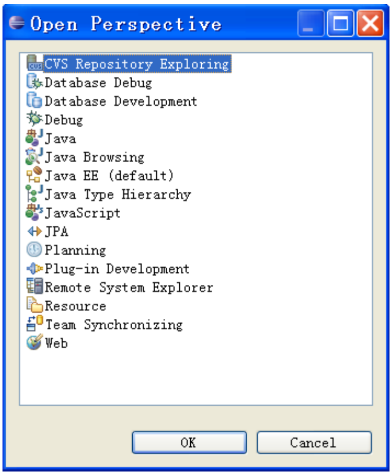


图 19-6　“Open Perspective”对话框

STEP 03 单击“OK”按钮。在 Eclipse 主窗体中，出现“CVS Repositories”视图。在该视图中任意位置单击右键，在弹出的菜单中，选择“New/ Repository Location…”，弹出“Add CVS Repository”对话框。

在“Host：”文本框中输入 CVSNT 服务器名称；在“Repository path：”文本框中输入“/CVSTest”；在“User：”文本框后输入“CVSUser1”；在“Password：”文本

框中输入对应密码；在“Connection type：”下拉列表中选择“pserver”。单击“Finish”按钮创建 Repository 位置信息，如图 19-7 所示。

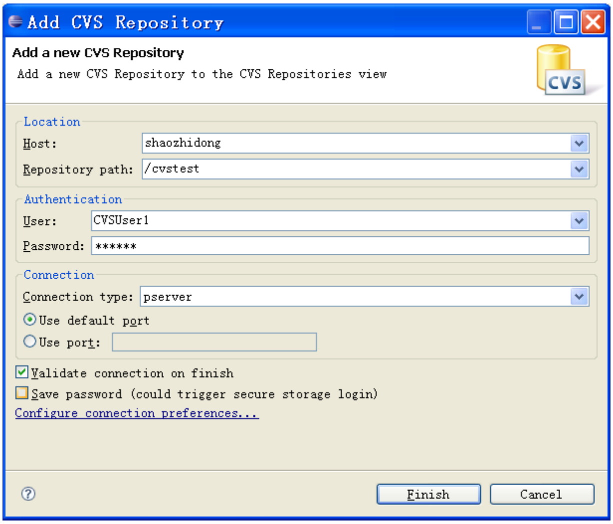


图 19-7 “Add CVS Repository”对话框

STEP 04 共享该工程。单击主菜单“Window\Show View\Project Explorer”切换到工程视图。在该视图中右键单击工程名“TestProject”，在弹出的菜单中，选择“Team→Share Project”，弹出“Share Project”对话框。选择刚才创建的 Repository 位置（也可以在此处新建），如图 19-8 所示。

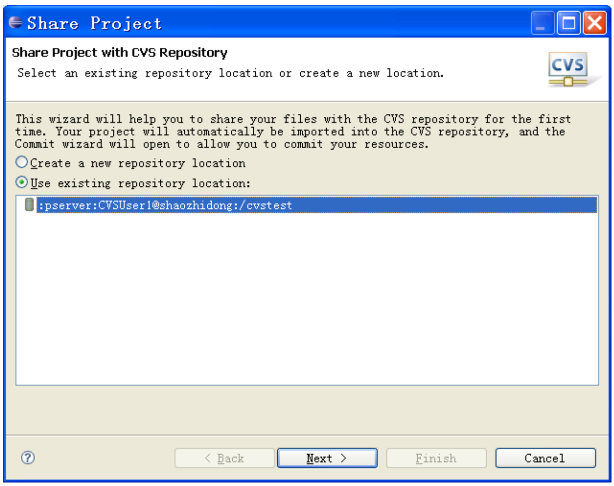


图 19-8 共享工程设置

STEP 05 单击“Next”，指定模块名。我们选择使用工程名作为模块名，如图 19-9 所示。

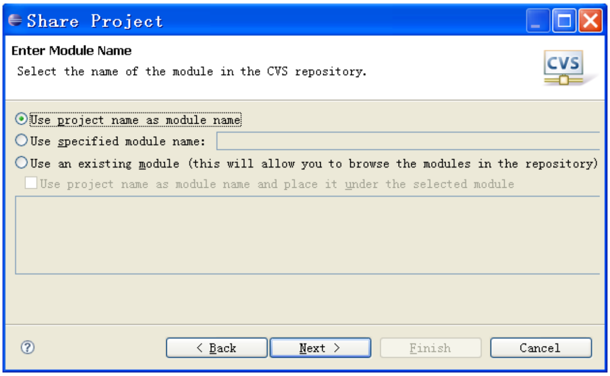


图 19-9 设置模块名称

STEP 06 单击“Next”按钮，进入下一页，显示工程中将要被提交和追加的文件。单击“Finish”按钮完成共享。弹出“Commit Files”对话框，输入注释“CVS 测试，共享创建！”，如图 19-10 所示。

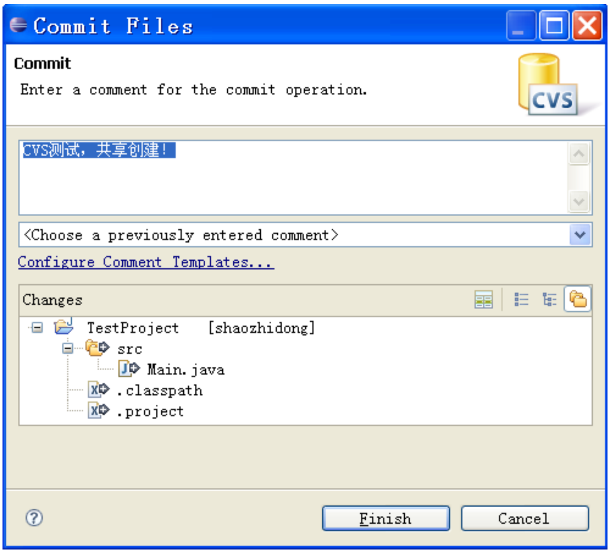


图 19-10 “提交”对话框

STEP 07 单击“Finish”确定。

STEP 08 打开 CVSNT 服务器上的 Windows 资源管理器，可以看到在指定的目录

下添加 TestProject 项，如图 19-11 所示。

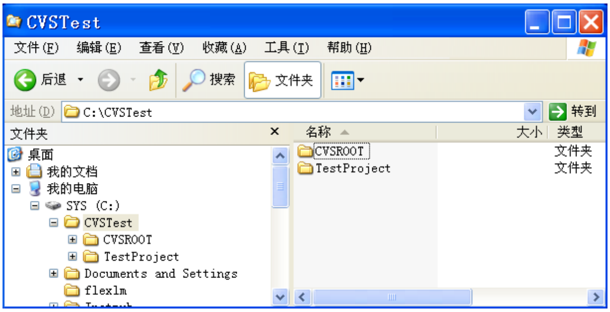


图 19-11 CVSNT 对应目录

19.4 在 Eclipse 客户端开发环境中使用 CVS

STEP 01 运行 Eclipse。选择“File→Import…”，弹出“Import”对话框，选择“CVS”下的“Project from CVS”，如图 19-12 所示。

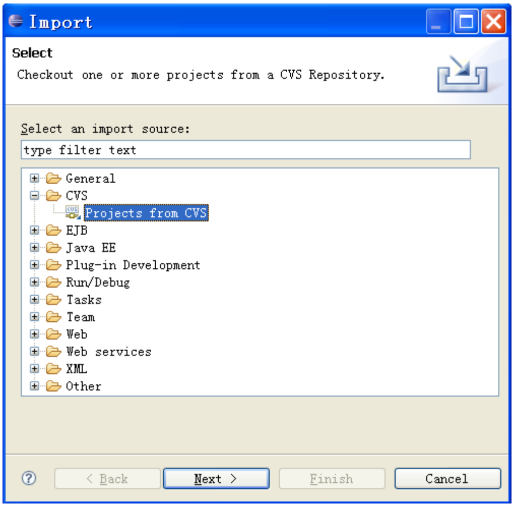


图 19-12 “Import”对话框

STEP 02 单击“Next”按钮。弹出“Checkout from CVS”对话框，如图 19-13 所示。

STEP 03 选择“Create a new repository location”项，创建新的 repository 项。具体步骤同上。输入服务器名、用户名、密码等创建新 Repository 所需要的信息后，单击“Next”，

进入“Select Module”对话框，选择“Use an existing module”项，如图 19-14 所示。

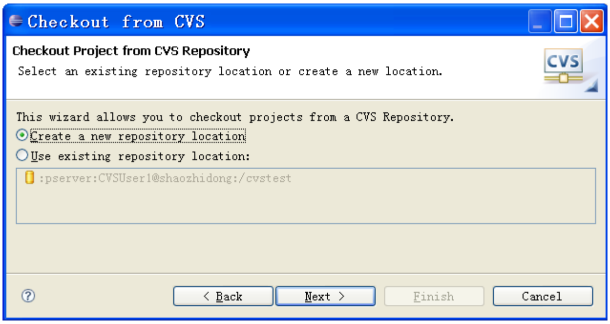


图 19-13　“Checkout from CVS”对话框

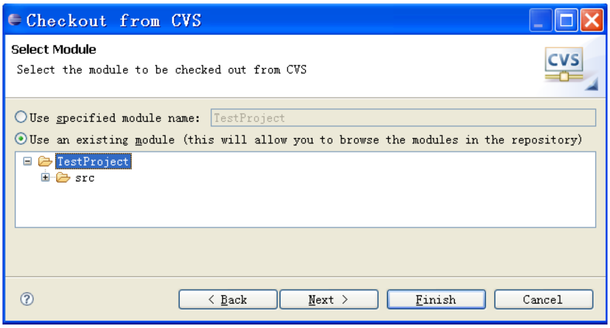


图 19-14　选择模块

STEP 04 单击“Next”，进入“Check Out As”对话框，如图 19-15 所示。

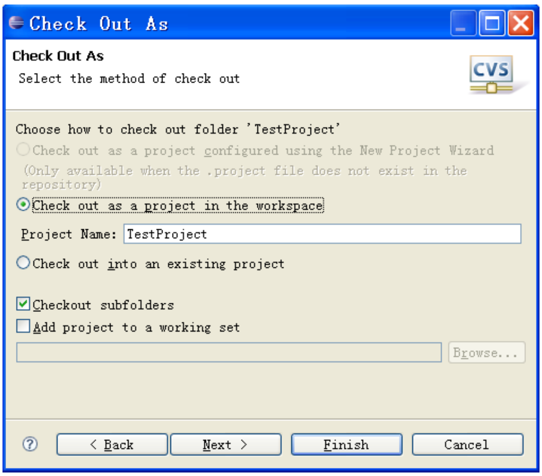


图 19-15　“Check Out As”对话框

STEP 05 单击“Next”，选择要保存的路径，如图 19-16 所示。

STEP 06 单击“Finish”，从服务器上读出文件，如图 19-17 所示。

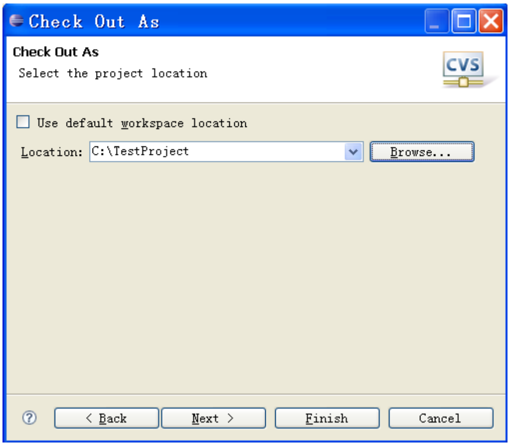


图 19-16　设置保存路径

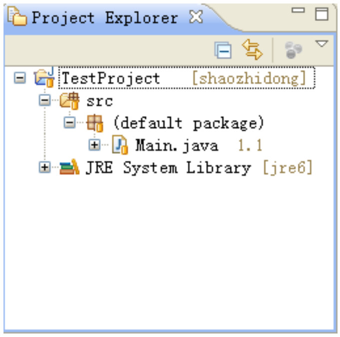


图 19-17　读出文件示意图

STEP 07 打开 Main.java 文件。单击主菜单“CVS→Show History”，可以查看修改历史，如图 19-18 所示。

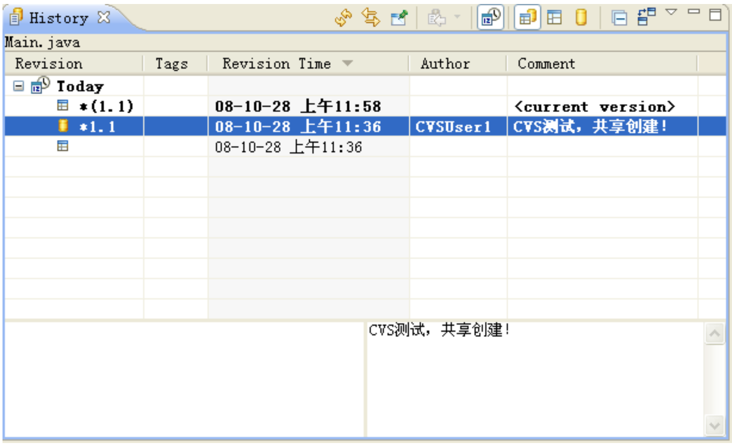


图 19-18　查看文件修改历史

STEP 08 在 main 函数中输入“System.out.print("Hello World");”，单击主菜单“CVS→Commit…”。在弹出的“Commit Files”对话框中，输入“修改主函数!”，如图 19-19 所示。

STEP 09 可以看到在项目视图中Main.java文件的版本发生了变化，如图19-20所示。

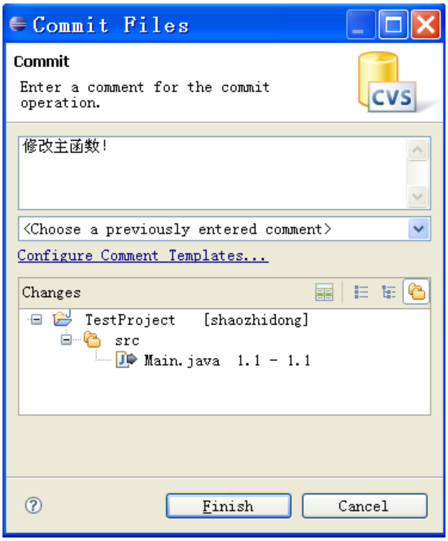


图 19-19 添加修改

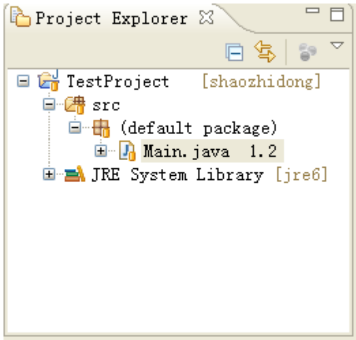


图 19-20 文件版本变化

STEP 10 通过再查看修改历史，可以看到修改记录，如图19-21所示。

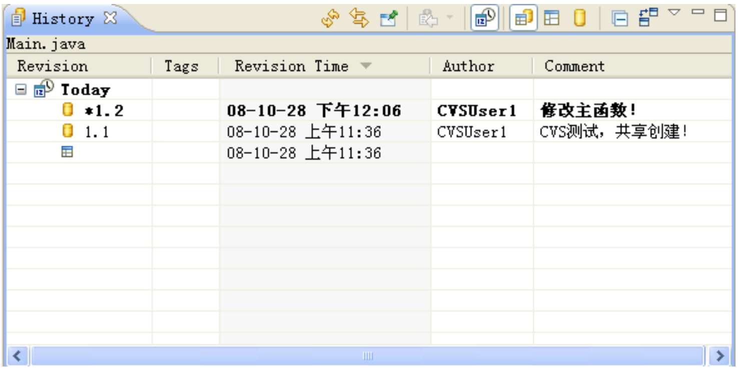


图 19-21 文件修改历史

19.5 VS.NET 中集成 CVS

可以通过使用插件来达到在 VS.NET 中使用 CVS 的目的。在使用之前，首先要按照 CVSNT。以下使用开发工具为 Visual Studio.NET 2005。

STEP 01 插件下载安装：可以使用 Jalindi Igloo 插件来完成上述功能，可以从以下网址下载该插件 http://www.jalindi.com/igloo/。

STEP 02 解压缩下载文件，单击目录下的 Setup.EXE 文件可以完成安装。

STEP 03 运行 VS.NET 开发环境，单击“工具”菜单上的“选项……”，弹出“选项”对话框。在对话框上左侧列表中，选择“源代码管理/插件选择”，选择“Jalindi Igloo”，如图 19-22 所示。

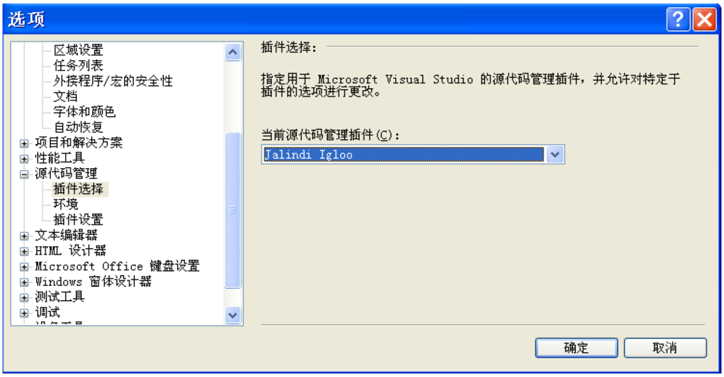


图 19-22 启动 Jalindi Igloo 插件

STEP 04 用 VS.NET 创建一个名为“CVSDemo”的 Windows 应用程序项目，如图 19-23 所示。

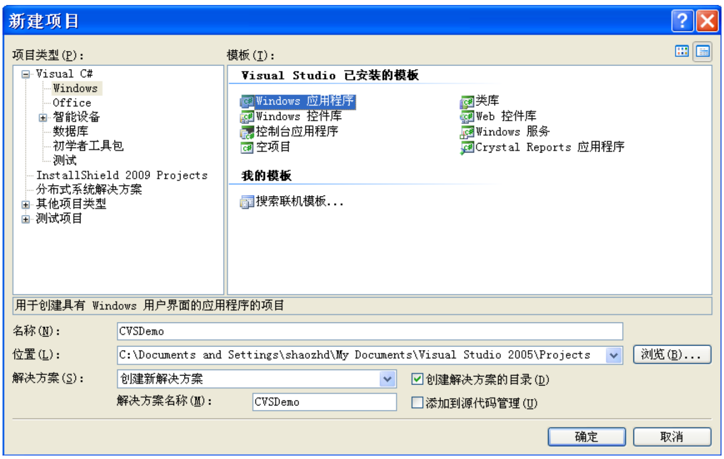


图 19-23 新建 Windows 项目

STEP 05 在 VS.NET“解决方案管理器”中选中当前解决方案，右键单击该解决方

案，在弹出的菜单中选择“将解决方案添加到源代码管理……”。弹出新的对话框。在“CVSROOT”文本框中输入“:pserver:CVSUser1@shaozhidong:/CVSTest”，如图 19-24 所示。

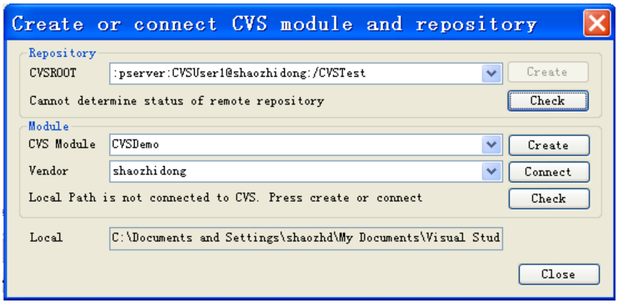


图 19-24　CVS 服务器登录设置

单击该文本框下方的“Check”测试连接，在弹出的对话框中输入 CVSUser1 对应的密码，如图 19-25 所示。



图 19-25　“CVS Login”对话框

CVS 的远程访问一般使用 pserver 协议，CVSROOT 的格式如下：

```
“:pserver:username:pasword@hostname:repository”
```

STEP 06 单击“Create”按钮创建新模块。创建完成后会弹出成功提示，如图 19-26 所示。



图 19-26　创建模块成功提示

STEP 07 在添加的过程中，对于一些临时文件、二进制文件的添加可能会报错，直接忽略就可以了。完成后可以看到“解决方案管理器”中添加了相关标识，如图 19-27 所示。

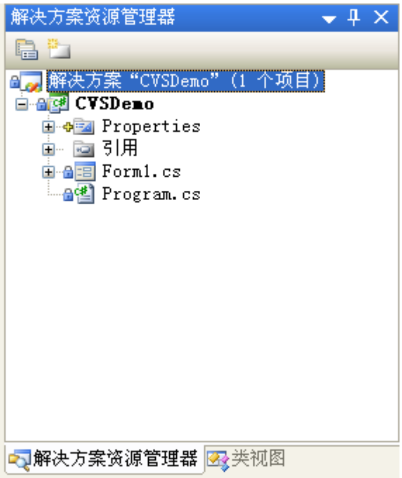


图 19-27 文件保护状态

STEP 08 以后的使用基本上和 VSS 的使用相同。具体可以参照前面 VSS 部分的介绍。

19.6 CVS 和 VSS 比较

- VSS 中的"签出（Check out）"和"恢复签出（Undo check out）"，在 CVS 中是"编辑（Edit）"和"恢复编辑（Unedit）"。
- VSS 中的"签入（Check in）"和"得到最新版本（Get last version）"，在 CVS 中为"Commit"和"Update"。
- CVS 没有文件锁定模式，VSS 在 check out 同时，记录了文件被导出者锁定。
- 像"$Header$"、"$Date$"这样的标记在 VSS 称之为 Keyword Explaination，默认是关闭的，需要通过 OPITION 打开，并指定需要进行源文件关键词扫描的文件类型：*.txt,*.java, *.html...；在 CVS 中，标记自动更新功能默认是打开的，因此不用-kb 方式添加 binary 文件的话在 cvs 自动更新时可能会导致文件失效。
- VSS 和 CVS 都通用的标记关键词有："$Header$"、"$Author$"、"$Date$"、"$Revision$"。

第 20 章　Visio 实例入门

Microsoft Visio 是最常用的画图工具。在本文中，我们将介绍如何用 Visio 绘制流程图。使用的软件是“Microsoft Visio 2007”。

该实例位于光盘的“DEMO\Visio\”目录下。

20.1 建立新文档

STEP 01 启动 Visio 软件。单击左侧类别中的“流程图”，如图 20-1 所示。

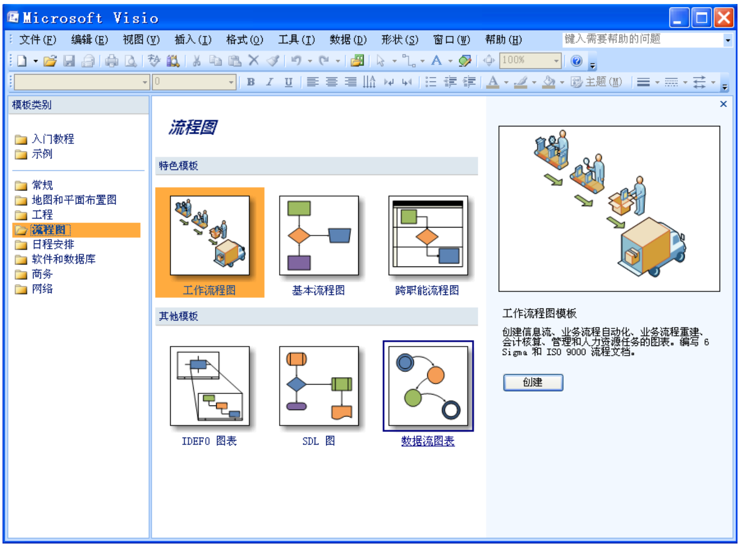


图 20-1　选择流程图样式

STEP 02 在“模板”区选择“基本流程图”图标，单击右面的“创建”按钮，进入“基本流程图制作窗口”，如图 20-2 所示。

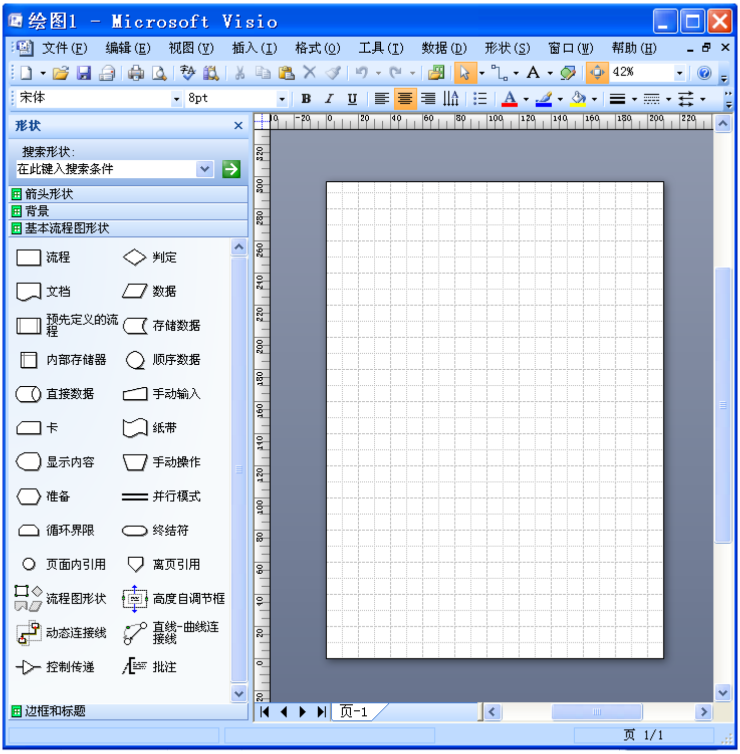


图 20-2　基本流程图制作窗口

20.2 创建流程图

STEP 01 从“基本流程图形状”选中“准备”图标，拖曳到绘图区。在绘图区，双击该形状，输入文字“开始”。在页边的空白处单击鼠标左键可以结束文字的输入。

STEP 02 单击鼠标左键，选择该图形，当形状周围显示绿色方块时，表示被选中了。然后，通过格式栏中的相应按钮对字体、字号和字的颜色等进行相应设置。这里将字体设置为宋体，字号为 14pt。

STEP 03 改变字的大小后，需要调整形状的大小。鼠标移到绿色方块处，当鼠标变为水平箭头时，按住拖动，可以调整形状的大小，以便让形状可以完全容纳文字。

STEP 04 同理添加“软件立项”、“需求分析”、“系统设计”、“文档”、“界面设计”、“建数据库”、“详细设计”、“编码”、“测试”、“提交客户”、“交付”、“项目经理”等形状，如图 20-3 所示。

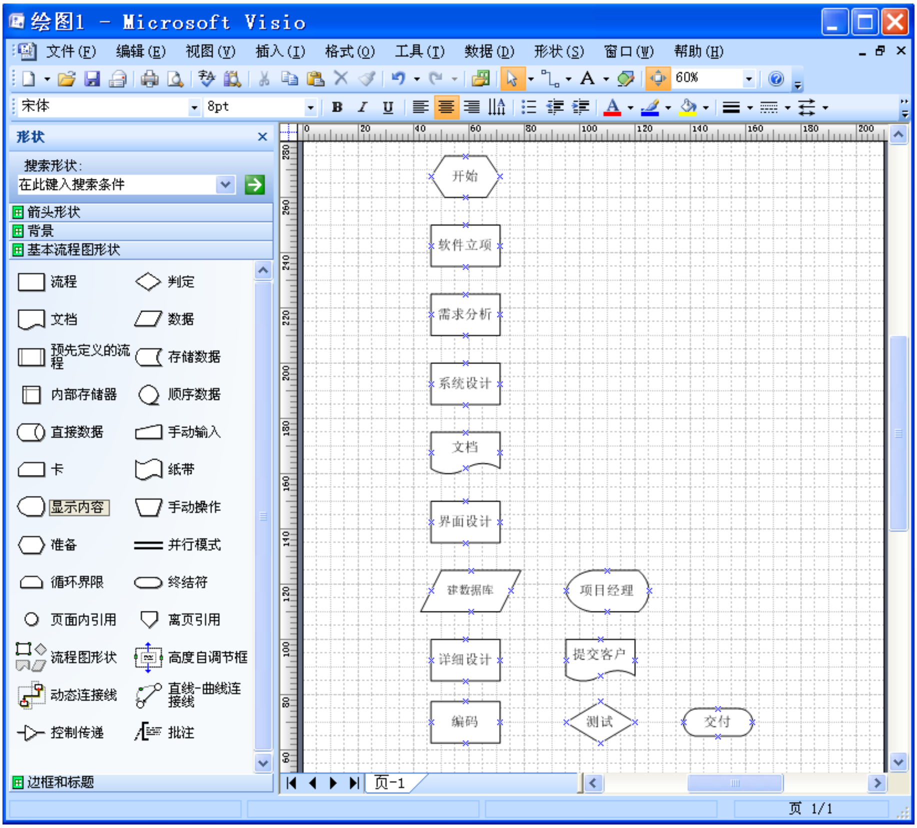


图 20-3 流程图制作

STEP 05 单击工具栏上的“连线工具”按钮，在弹出的下拉菜单中选择“连接线工具”项，如图 20-4 所示。

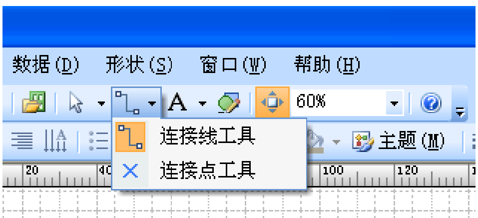


图 20-4 “连接线工具”

STEP 06 返回绘图区域后，将鼠标移至“开始”图示下方的节点位置，按住鼠标左键并向下拖拉至“软件立项”图示框上节点位置，即可在两个图示框之间建立带方向的箭头连线。不需要连接线的时候，单击“指针工具”，回到默认状态。

STEP 07 同理连接其他形状，如图 20-5 所示。

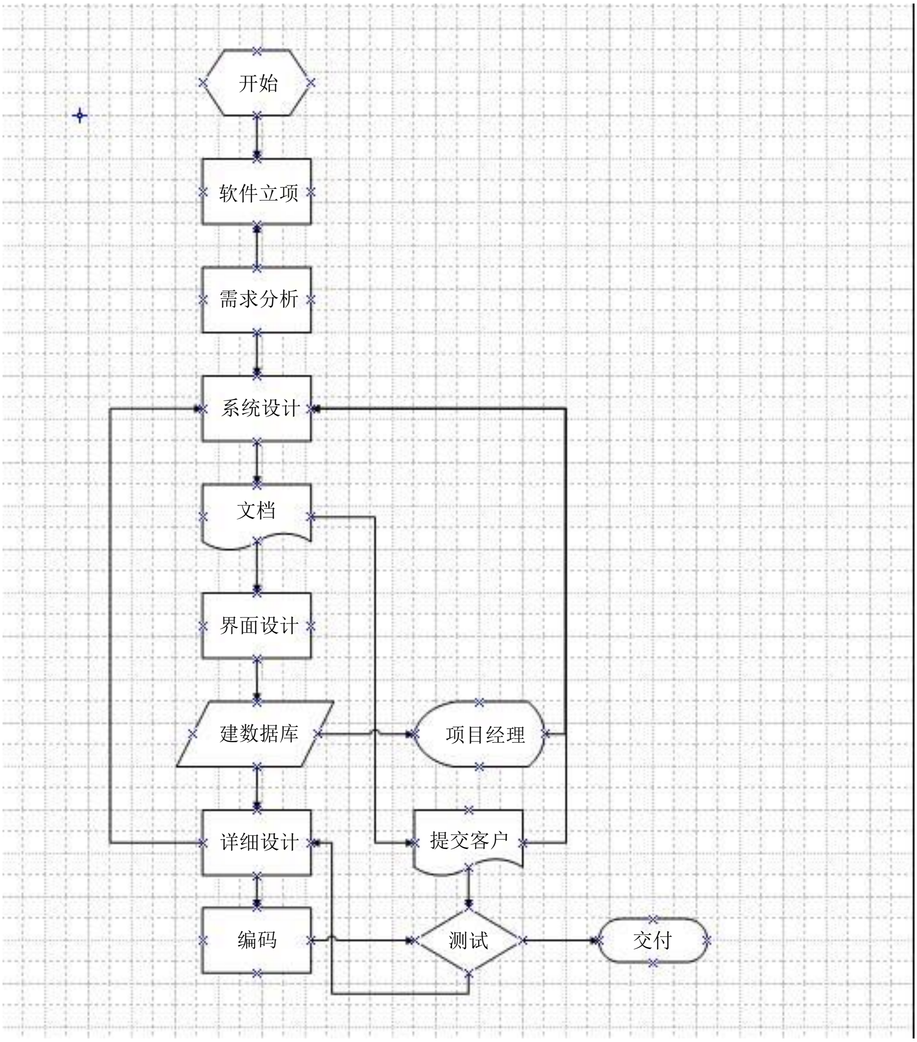


图 20-5 流程图

STEP 08 在“基本流程图形状”区用鼠标拖拉“批注”图标至“软件立项”框右侧，以添加批注说明内容。双击批注框，输入内容“要开立项评审会议”按回车键。

STEP 09 接下来添加连接线文字。双击“测试”与“详细设计”两个图示框的连接线，出现文本输入状态，在文本框中输入“未通过”，表示如果测试未通过，则返回“详细设计”过程，输入完成后单击框外任意区域可结束输入。

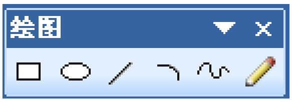
STEP 10 为了美化流程图，还可以为它添加背景图案。单击“形状”区的“背景”标题栏，在它下方的框内随即显示了背景模板图示。选择“轨道背景”并拖拉至右侧绘图区，就为流程图配置好背景图案了。

20.3 保存

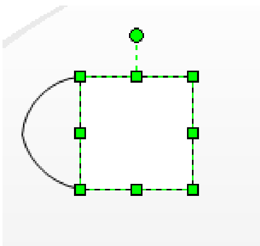
完成所有的操作之后，单击菜单“文件→保存”，在“另存为”对话框中的“文件名”文本框里输入“软件开发流程图”，然后设置好保存路径，单击“确定”按钮就完成任务了。

20.4 自定义形状的制作

STEP 01 在工具栏空白处右键选择“绘图”，出现“绘图”工具条，选择“弧线工具”。



STEP 02 先选择起点画出半个圆弧，然后再画出另外半个圆弧，使其刚好对称，然后再画一个矩形。



STEP 03 回到“指针”，选中两个半弧以及矩形，单击右键，在出现的菜单中选择“形状→组合”，使这个形状成为一个整体。

STEP 04 选中该形状。选择主菜单“文件→形状→新建模具”。

STEP 05 右键单击新建的模具，在弹出的菜单中选择“属性”。在弹出的对话框中，把新建的模具重新命名为“我的模具”，如图 20-6 所示。

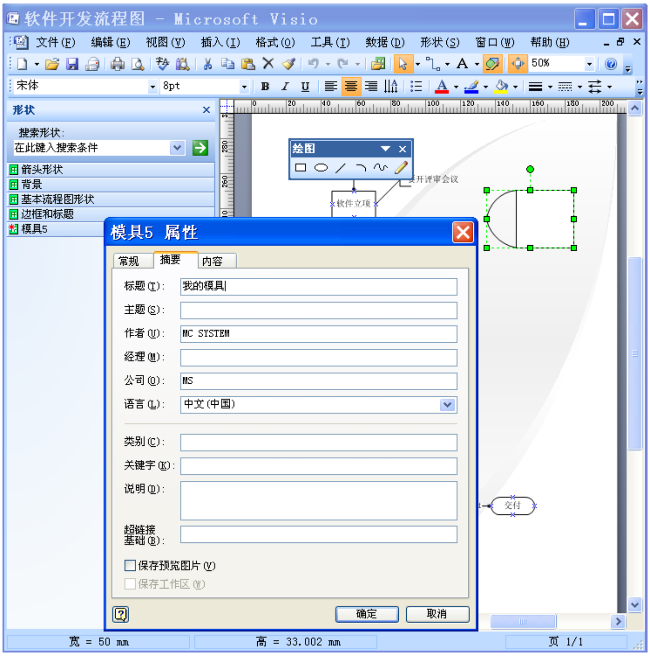


图 20-6　设置模具属性

STEP 06 把刚才创建的组合形状拖曳回到“我的模具”中，双击该图标重新命名为“组合形状”，如图 20-7 所示。

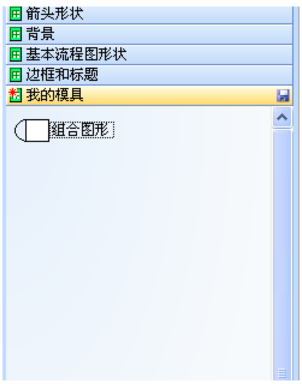


图 20-7　组合形状

STEP 07 在新建的模具“我的模具”位置单击右键，选择“另存为……”。弹出“另存为”对话框。“我的形状”，注意保存类型并改名，如图 20-8 所示。

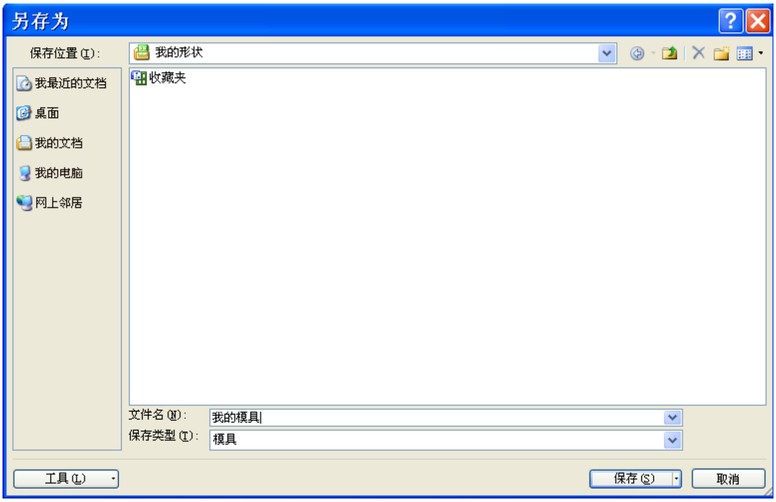


图 20-8 “模具另存为”对话框

STEP 08 这个模具在“形状”里的“我的形状”里可以看到，以后使用起来也很方便，如图 20-9 所示。

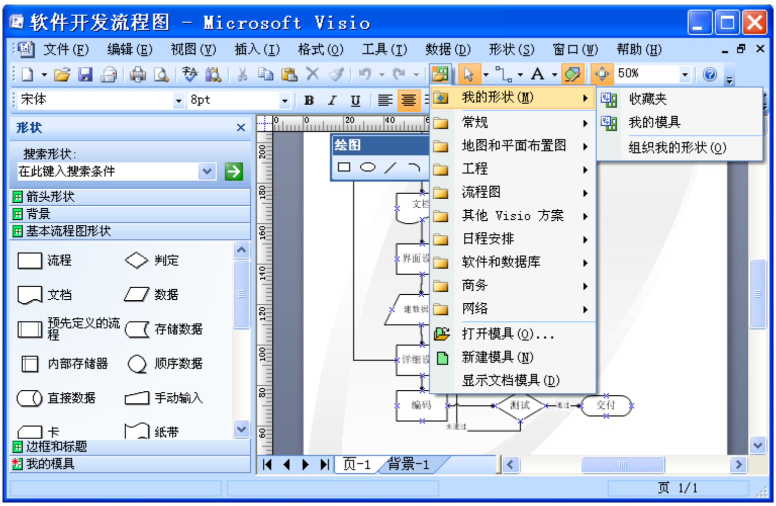


图 20-9 我的形状菜单

第3篇 实例篇

本篇导读

本篇中，我们详细介绍了一个电子商务系统——“网络商城系统”的开发过程。从需求分析到软件加密、安装盘制作等产品化，每个环节都进行了详细的讲述，有非常实用的价值。

在本书的配盘中，我们提供了本项目的所有开发文档，包括源代码、数据库设计文档、安装盘源文件等，以方便大家的学习。

第 21 章　网上商城系统开发实战

21.1 项目概述

随着各种网络技术的发展，Internet 对人类生活的影响越来越大，电子商务就是其中最典型的一个应用。

电子商务，Electronic Commerce，通常是指在全球各地广泛的商业贸易活动中，在因特网开放的网络环境下，基于浏览器/服务器应用方式，买卖双方不谋面地进行各种商贸活动，实现消费者的网上购物、商户之间的网上交易和在线电子支付，以及各种商务活动、交易活动、金融活动和相关的综合服务活动的一种新型的商业运营模式。电子商务网站的成立是当今 e 时代方便人们生活的又一伟大创举。在生活节奏越来越快的今天，人们需要有更多的休息时间，而网上购物给人们带来轻松，让每个人享受购物的快乐和冲浪的乐趣。

电子商务涵盖的范围很广，一般可分为企业对企业（Business-to-Business），或企业对消费者（Business-to-Consumer）两种。另外还有消费者对消费者（Consumer-to-Consumer）这种大步增长的模式。随着国内 Internet 使用人口的增加和各大银行对网络支付方式的推进，利用 Internet 进行网络购物并以银行卡付款的消费方式已渐流行，市场份额也在快速增长，电子商务网站也层出不穷。

目前，在 Internet 上遍布了各种类型的 B2C 网站，提供从鲜花、书籍到计算机、汽车等各种消费品和服务。比如国际的亚马逊和国内的当当网就是非常典型的 B2C 例子。

网上购物系统，是在网络上建立一个虚拟的购物商场，避免了挑选商品的烦琐过程，使用户的购物过程变得轻松、快捷、方便，很适合现代人快节奏的生活；同时又能有效

地控制“商场”运营的成本，开辟了一个新的销售渠道。

建立网上购物商城，帮助企业提高生产效率，降低经营成本，优化资源配置，从而实现社会财富的最大化的实用网络应用系统。

21.2 可行性分析

21.2.1 系统目标

网上商城系统是基于 B/S 模式，通过对一些典型电子商务网站的考察、分析，并结合企业要求而开发的一套拥有前后台的信息化管理系统。

本系统计划达到以下目标：

- 为顾客提供一个网络服务平台：网络服务平台就是顾客能打破时空界限，能够通过网络实现从浏览商品到下订单的整个流程。
- 为工作人员提供一个工作平台：工作平台就是员工直接在网络上完成日常事务，是实现无纸化办公的反映，即职工通过计算机完成采集信息，处理信息，分析信息等工作。工作平台是提高工作效率的基础。
- 为管理者提供一个控制平台：控制平台就是管理者能通过业务控制平台，即把企业的各项制度、标准，通过程序控制落实到企业各项工作活动中去。通过对工作流进行设置与监控，从而能严格控制企业活动的各项动作，实现事务的有效管理。
- 为系统维护者提供一个集中维护平台：集中维护平台就是系统管理人员能够对系统运行的缺陷、故障进行集中处理，使系统管理人员能够快速、有效、连续地对系统进行维护与调整。

21.2.2 开发的可行性分析

1．应用上的可行性

随着企业信息化的推进，各个企业对电子商务的应用逐渐认可，并且产生了很大的热情。对企业运营商来说，建立电子商务网站，也是企业对外展示信息，从事商务活动的窗口和界面。

2．技术上的可行性

随着网络编程技术的发展，特别是 Java 技术、微软.NET 技术的出现，编写网络应用程序的难度逐渐降低；同时网络硬件设备的不断更新换代，也使得原先阻碍网络应用程序发展的性能问题不再存在；各种网络安全技术的出现和更新，也使得电子商务的应用更加普及。这些都保证了整个系统开发从技术上是可行的。

3．经济上的可行性

电子商务网站最大的特点就是成本低廉。它不受时间和空间的限制，甚至可以做到以需订购，可以减低库存，减少管理成本，从而使得企业的运营更加主动、更加灵活。

21.3 需求分析

正如前面“知识篇”所述，可行性分析是决定“做还是不做？”，而用户需求分析阶段是决定“做什么，不做什么”。

对于一般系统，笔者认为，在用户需求分析阶段主要要思考以下三个核心问题：

- 系统有哪些用户？
- 这些用户的工作流程是什么？
- 本软件当前版本要实现哪些功能？以后要实现的功能有哪些？

21.3.1 确定用户类型

很显然，对于网上商城，应该共有两大类用户：

- 普通顾客，他们要通过商城系统购买物品。
- 工作人员：要处理订单、管理商品信息、一些系统消息，甚至要处理用户的投诉信息。

21.3.2 用户工作流程

顾客的购物流程如图 21-1 所示。

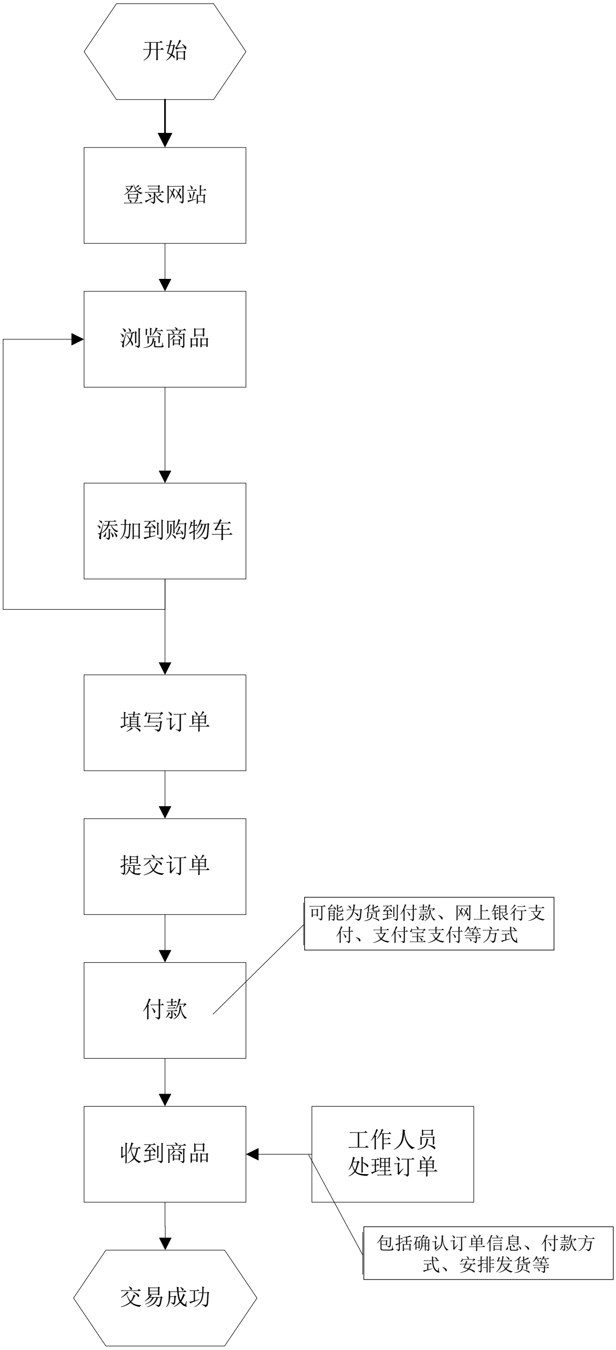


图 21-1　顾客的购物流程图

工作人员的工作流程如图 21-2 所示。

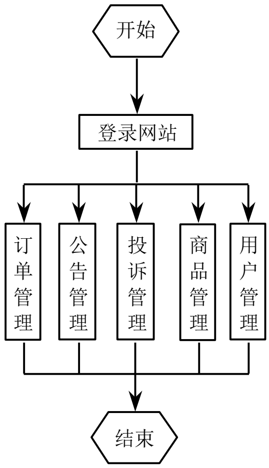


图 21-2　工作人员的工作流程图

完成以上方面的工作后，接下来我们开始进行需求分析阶段最重要的工作，即决定系统要实现哪些功能。

21.3.3　软件功能

1．顾客购物

为了实现顾客购物流程，系统需要完成以下功能：

- 顾客浏览商品
- 查询商品
- 订购商品
- 购物车
- 用户信息维护等功能（用户注册、信息修改）

2．工作人员管理

为了实现工作人员的工作流程，系统需要完成以下功能：

- 公告管理：包括添加商城公告、修改公告、删除公告。
- 商品管理：包括添加商品类别、修改商品类别、删除商品类别、添加商品信息（类

别、名称、编号、所属公司等）、商品图片上传、修改、删除、修改商品信息、删除商品信息、查看商品信息等。

- 订单管理：包括处理订单、办理发货、办理结账、删除订单等。
- 投拆管理：包括录入投拆的解决方法、删除已经解决的投拆、查看投拆用户等。
- 系统用户管理功能：包括添加系统用户、修改系统用户信息、删除系统用户信息。

3. 功能示意图

我们把顾客购物模块称为“前台”模块；工作人员管理模块称为“后台管理”模块。图 21-3 是整个系统的功能示意图。

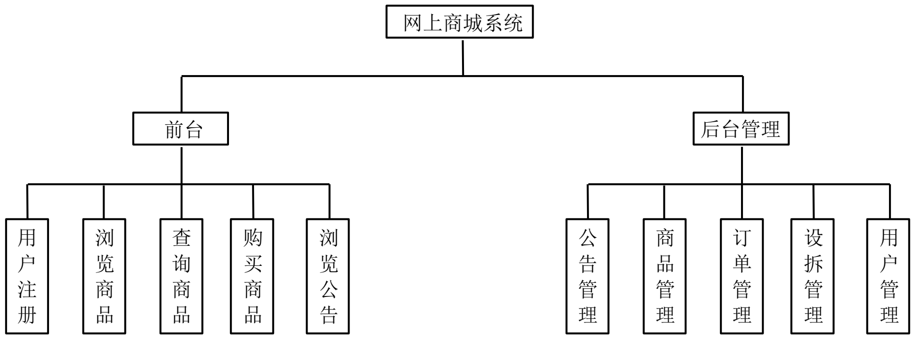


图 21-3　系统的功能示意图

4. 以后实现的功能

- 用户权限

工作人员按照权限不同，也可以分为不同的类型，比如公告管理和订单管理分别都有指定类型人员处理，而具有最高权限的用户才拥有用户管理功能。

本系统该版本中暂定不实现这个功能。即把工作人员当作一类用户对待，他拥有以上全部功能。

- 用户投诉

用户投诉功能在本系统中暂不实现，读者可以参照已经完成的功能自行开发。

21.3.4　UML 图

对照以上思路，我们在 Rational Rose 中建立模型图如图 21-4 所示。

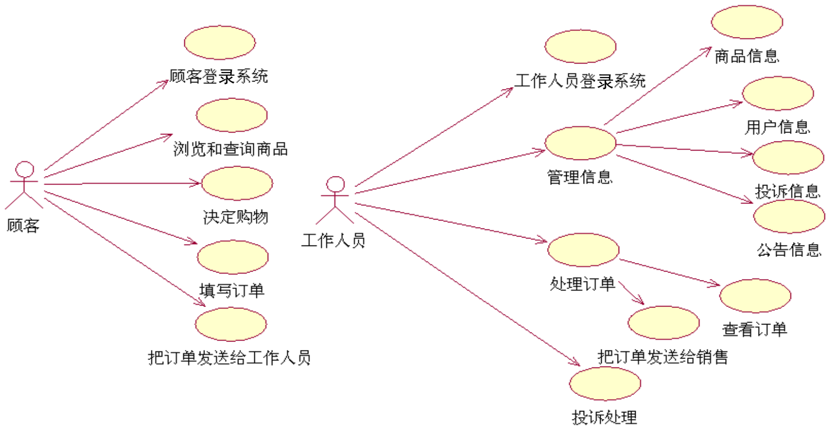


图 21-4　选题 UML 图

在本书附带光盘中“Demo\Project\Rose 图”目录下找到该文件，名字为“OnlineShop.mdl”。

21.3.5　系统运行环境

1．开发环境

本系统计划使用 VisualStudio.NET 2005 作为开发工具，编程语言是 C#，编程技术是 ASP.NET。

使用 VS.NET 的最大优点是容易上手，开发效率高，很多框架代码都由 VS.NET 自动完成。

使用 C#，是因为它是微软为.NET 平台量身定做的程序语言，既拥有 C++的强大的功能又有 Visual Basic 简易的特性，它在带来对应用程序的快速开发能力的同时，并没有牺牲 C 与 C++程序员所关心的各种特性，C#忠实地继承了 C 和 C++的优点。可以说，C#是最能够体现.NET 优势的语言。

相对于 ASP 来说，ASP.NET 最大的特点是编译执行，而不是 ASP 的解释执行，这样使得整个程序的执行效率大大增加。同时由于 ASP.NET 是建立在.NET 框架之上的，因此.NET 的各种优点都可以使用。同时由于采用了 CodeBehind 技术，它的程序代码与页面标识可以完全剥离，不会混合在一个页面上，这样便于程序员修改。

注意：读者也可以参照本书设计，用 Java 技术来实现网上商城系统。

2．数据库

由于使用了 VisualStudio.NET 作为开发工具，考虑到数据库的安全性和稳定性，后台数据库同样还是采用微软的产品 SQL Server 2005 Express。使用 SQL Server 2005 Express 的原因是易于上手，安装简便，并且是免费产品，同时又具有 SQL Server 2005 的一些新特性。它能与 VS2005 中的项目无缝集成，且这种数据库还有一个好处就是，只需要把数据库文件包含在项目中就可以使用。

Microsoft SQL Server Management Studio Express（SSMSE），是一种免费、易用的图形管理工具，用于管理 SQL Server 2005 Express Edition。

Microsoft SQL Server2005 Express 和 SQL Server Management Studio Express 都可以在微软网站上下载到（http://www.microsoft.com/downloads/）。

在 ASP.NET 中使用 ADO.NET（ActiveX Data Objects）访问数据库。ADO.NET 是基于 Microsoft 公司推出的.NET 框架中的用于数据访问的组件。

3．运行环境

由于使用了以上开发技术，本系统服务器的运行环境如下：

- 操作系统：Windows XP 以上
- 服务器：IIS5.0 以上
- 浏览器：IE6.0
- 数据库：SQL Server 2005 Express
- 硬件配置：CPU：P41.8GHz
- 内存：256MB 以上
- 分辨率：最佳效果 1024×768 像素
- 需安装.NET Framework2.0

21.4 系统设计

结合前面知识篇所述，对于一般 B/S 结构系统，笔者认为，在系统设计阶段主要进行以下两个方面的设计：

1．数据库设计

- 要设计几张表？
- 各个表有哪几个字段？
- 每个表之间的关系？

2．页面设计

- 需要几个页面？
- 各个页面布局？
- 各个页面之间的跳转关系？（即要把用户的工作流程转换为系统的执行流程。）

21.4.1　数据库设计

数据库设计要遵循一些规则，一个好的数据库满足一些严格的约束和要求。尽量分离各实体对应的表，一个实体对应一个表，搞清楚该实体有哪些属性，对应有些什么字段，以及各实体之间有何种联系。实体、属性与联系是进行系统设计时要考虑的 3 个元素，也是一个好的数据库设计的核心。

1．实体关系图

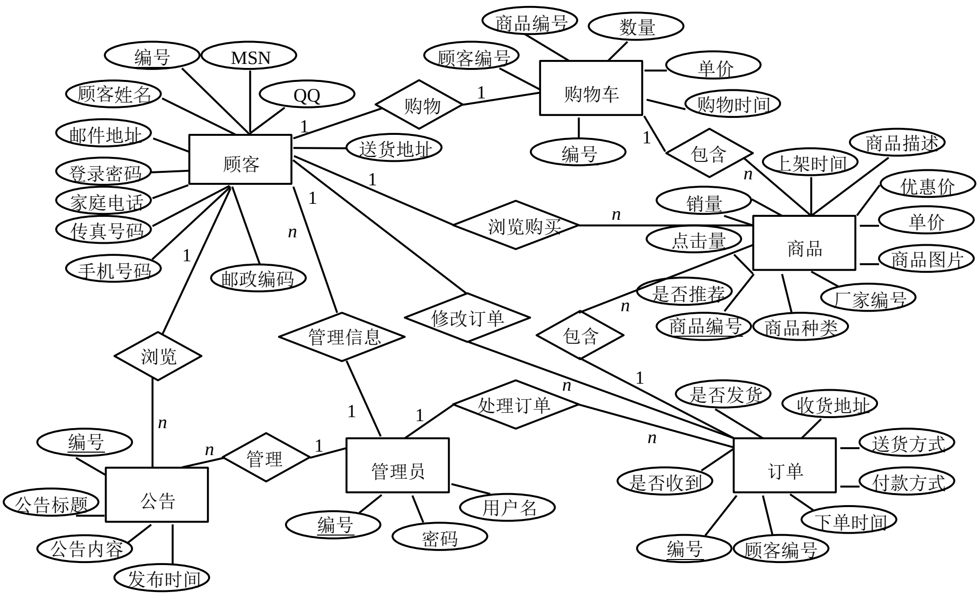


图 21-5　实体关系图

2．表结构

网上购物系统中主要涉及的实体有“管理员”、“顾客”、“商品”、“购物车”、“公告”、“订单”。

- 网上购物系统的主要对象是顾客，所以必须建立**顾客表**，包括顾客的基本信息情况。
- 用户的主要活动是对商品的购买，所以必须建立**商品信息表**。
- 用户对商品的购买，选择所要的商品，首先得对商品进行搜索，所以我们把商品进行分类，建立**商品类别表**。
- 用户要购物，因此建立**购物车表**。
- 用户提交定单，建立**订单表**。
- 每个订单要包含多个购买的商品，因此要建立**订单详情表**。
- 顾客可以浏览商城的公告信息，因此要建立**公告信息表**。
- 另外有管理员进行信息处理，因此要有**管理员表**。

经过分析，此数据库包含如下表：

- 顾客表：Customers
- 商品分类表：Categories
- 商品表：Products
- 订单表：Orders
- 订单详情表：OrderDetails
- 购物车表：Shoppingcart
- 公告表：News
- 管理员表：Managers

具体表说明如下：

Customers 用户表

列　　名	说　　明	类型（长度）	备　　注
ID	顾客 id	Int	不允许空，主键

续表

列　　名	说　　明	类型（长度）	备　　注
Name	顾客名称	Varchar (50)	不允许空
Password	登录密码	Varchar (50)	不允许空
Phone	家庭电话	Varchar (50)	允许空
Fax	传真号码	Varchar (50)	允许空
MobilePhone	移动电话	Varchar (50)	允许空
E-mail	邮件地址	Varchar (100)	允许空
Address	详细地址	Varchar (255)	允许空
Postcode	邮政编码	Char(6)	允许空
QQ	QQ 号	Varchar (50)	允许空
MSN	MSN 号	Varchar (50)	允许空

Categories 商品分类表

列　　名	说　　明	类型（长度）	备　　注
ID	商品类型 id	Int (4)	不允许空，主键
Name	商品类型名称	Varchar (50)	不允许空
Memo	商品类型描述	Varchar (255)	允许空

Products 商品表

列　　名	说　　明	类型（长度）	备　　注
ProductID	商品 ID	int	不允许为空
CategoryID	分类 ID	int	不允许为空
ModelNumber	商品编号	varchar(50)	允许为空
ModelName	商品名称	varchar(255)	允许为空
ProductImage	商品图片文件名	varchar(50)	允许为空
UnitCost	单价	money	不允许为空
Discount	折扣	money	不允许为空
Description	描述	varchar(2000)	允许为空
DateTimed	上架时间	smalldatetime	不允许为空
Clicks	点击量	int	不允许为空
Sales	销售量	int	不允许为空
Commend	是否推荐	bit	不允许为空

Orders 订单表

列　　名	说　　明	类型（长度）	备　　注
ID	订单 ID	int	不允许为空

续表

列　　名	说　　明	类型（长度）	备　　注
Memo	备注	varchar(2000)	允许为空
OrderDate	下订单时间	smalldatetime	不允许为空
CustomerID	用户 ID	int	不允许为空
ShipDate	发货时间	smalldatetime	不允许为空
HasSend	是否送货	bit	不允许为空
HasReceive	顾客是否收到	bit	不允许为空
PayWay	付款方式	varchar(50)	允许为空
CarryWay	送货方式	varchar(50)	允许为空
ReceiveAddress	收货地址	varchar(2000)	允许为空

OrderDetails 订单详情表

列　　名	说　　明	类型（长度）	备　　注
ID	ID	Int (4)	不允许空，主键
OrderID	订单 ID	Int (4)	不允许空
ProductID	商品 ID	Int (4)	不允许空
Quantity	商品数量	Int (4)	不允许空
UnitCost	商品单价	money	允许空

Shoppingcart 购物车表

列　　名	说　　明	类型（长度）	备　　注
ID	ID	int	不允许为空
CartID	购物车 ID	varchar(50)	不允许为空
ProductID	商品 ID	int	不允许为空
Quantity	商品数量	int	不允许为空
DateTimed	购买时间	smalldatetime	允许为空

News 公告表

列　　名	说　　明	类型（长度）	备　　注
ID	公告 ID	int	不允许为空
Name	公告标题	varchar(50)	允许为空
Memo	公告内容	varchar(255)	允许为空
DateTimed	公告时间	smalldatetime	不允许为空

Managers 管理员表

列　　名	说　　明	类型（长度）	备　　注
ID	管理员主键	int	不允许为空
Name	管理员姓名	varchar(50)	不允许为空
Password	管理员密码	varchar(50)	允许为空

3．存储过程

设计好数据库后，程序中主要的操作就是对以上的表进行“查询、添加、删除、修改”操作，因此对每张表都要进行以上四种操作的设计。

为了方便对数据库的操作，防止 SQL 注入攻击，提高安全性，我们用存储过程的方式来设计数据库操作。以下是程序中用到的存储过程一览表：

序号	操作对象	存储过程名称	功能描述	参数说明
1	Categories 表	spCategoryList	查询商品类型	无参数
2		spCategoryDelete	删除指定商品类型	@ID：要删除的商品种类 ID
3		spCategoryInsert	添加新商品类型	@Name：商品类型名 @Memo：商品类型说明 @ID：输出参数，返回新生成的商品类型 ID
4		spCategoryUpate	更新商品类型	@Name：商品类型名 @Memo：商品类型说明 @ID：要更新的类型 ID
5	Customers 表	spCustomerList	查询顾客信息	无参数
6		spCustomerDelete	删除指定用户信息	@ID：要删除的顾客 ID
7		spCustomerInsert	添加新用户	@Name：顾客名称 @Password：密码 @Phone：家庭电话 @Fax：传真号码 @MobilePhone：手机号 @E-mail：邮件地址 @Address：联系地址 @Postcode：邮政编码 @QQ：QQ 号码 @MSN：MSN 号码 @ID：输出参数，返回新生成的顾客 ID

续表

序号	操作对象	存储过程名称	功能描述	参数说明
8		spCustomerUpate	更新顾客信息	@Name：顾客名称 @Password：密码 @Phone：家庭电话 @Fax：传真号码 @MobilePhone：手机号 @E-mail：邮件地址 @Address：联系地址 @Postcode：邮政编码 @QQ：QQ 号码 @MSN：MSN 号码 @ID：顾客 ID
9		spCustomersLogin	顾客登录	@E-mail：用户 E-mail @Password：用户密码 @ID：输出参数，登录成功返回该顾客 ID，否则返回-1
10		spCustomersUpdatePassword	更新顾客密码	@ID：要更新密码的用户 ID @PasswordOld：原密码 @PasswordNew：新密码 @Result：输出参数，返回更新是否成功字符串
11	Managers 表	spManagerList	查询管理员信息	无参数
12		spManagerDelete	删除指定管理员账号	@ID：要删除的管理员账号 ID
13		spManagerInsert	添加新管理员	@Name：管理员登录名 @Pass：密码 @ID：输出参数，返回新生成的管理员用户 ID
14		spManagerUpate	更新管理员信息	@Name：管理员登录名 @Pass：密码 @ID：要更新的管理员账号 ID
15		spManagerLogin	管理员登录	@Name：登录名称 @Password：密码 @ID：输出参数，成功返回管理员 ID，失败返回-1

续表

序号	操作对象	存储过程名称	功能描述	参数说明
16	News 表	spNewsList	查询公告	无参数
17		spNewsDelete	删除指定公告	@ID：要删除的公告 ID
18		spNewsInsert	添加新商品类型	@Name：公告主题 @Memo：公告内容 @DateTimed：公告发布时间 @ID：输出参数，返回新生成的商公告 ID
19		spNewsUpate	更新公告信息	@Name：公告主题 @Memo：公告内容 @DateTimed：公告发布时间 @ID：要更新的公告 ID
20	Orders 表	spOrderList	查询订单	无参数
21		spOrdersDelete	删除指定订单，同时删除相应订单详情记录	@ID：要删除的订单 ID
22		spOrdersInsert	添加新订单，同时添加订单详情表	@CartID：购物车 ID @CustomerID：用户 ID @OrderDate：购买时间 @PayWay：付款方式 @CarryWay：送货方式 @Memo：备注 @ReceiveAdress：收货地址 @ID：输出参数，返回新生成的订单 ID
23		spOrdersAdminUpate	管理员处理订单	@ID：要更新的订单 ID @HasSend：是否送货 @HasReceive：是否收到
24		spOrdersUserUpate	用户修改订单	@ID：要修改的订单 ID @PayWay：付款方式 @CarryWay：送货方式 @ReceiveAdress：送货地址 @Memo：备注
25	Products 表	spProductList	查询商品信息	无参数
26		spProductsWhereCommend	查询推荐商品信息	无参数
27		SpProductDelete	删除指定商品	@ProductID：要删除的商品 ID

续表

序号	操作对象	存储过程名称	功能描述	参数说明
28	Products 表	spProductInsert	添加新商品	@CategoryID：商量类型 ID @ModelNumber：商品编号 @ModelName：商品名称 @ProductImage：商品图片 @UnitCost：单价 @Discount：优惠价 @DateTimed：上架时间 @Clicks：点击量 @Sales：销售量 @Commend：是否推荐 @ProductID：输出参数，返回新生成的商品 ID
29		spProductUpate	更新商品信息	@ProductID：要更新的商品 ID @CategoryID：商量类型 ID @ModelNumber：商品编号 @ModelName：商品名称 @ProductImage：商品图片 @UnitCost：单价 @Discount：优惠价 @DateTimed：上架时间 @Clicks：点击量 @Sales：销售量 @Commend：是否推荐
30	ShoppingCart 表	spShoppingCartList	查询指定购物车	@CartID：购物车 ID
31		spShoppingCartEmpty	清空指定的购物车	@CartId：购物车 ID
32		spShoppingCartAddItem	向购物车添加商品	@CartID：购物车 IT @ProductID：商品 ID @Quantity：商品数量
33		spShoppingCartRemoveItem	删除购物车中指定商品	@CartID：购物车 ID @ProductID：商品 ID
34		spShoppingCartTotal	计算指定购物车内商品总价格	@CartID：购物车 ID @TotalCost：输出参数，返回商品总价

续表

序号	操作对象	存储过程名称	功能描述	参数说明
35	ShoppingCart 表	spShoppingCartUpdateQty	更新购物车内指定商品数量	@CartID：购物车 ID @ProductID：商品 ID @Quantity：数量
36		spShoppingCartMigrate	更新购物车 ID	@OldCartID：旧 ID @NewCartID：新 ID
37	OrderDetails 表	spOrderDetailList	查询指定订单的详细情况	@OrderID:订单号

对数据库表的操作一般就是“查询、添加、删除、修改”四个操作，以下 4 个存储过程说明了对顾客表 Customers 的这 4 个操作：

1．查询：spCustomerList

```
ALTER PROCEDURE dbo.spCustomerList

AS
 SELECT * FROM dbo.Customers
```

2．添加：spCustomerInsert（其中用到了事务处理）

```
ALTER PROCEDURE dbo.spCustomerInsert
(
   @Name      VARCHAR(50),
   @Password   VARCHAR(50),
   @Phone     VARCHAR(50),
   @Fax       VARCHAR(50),
   @MobilePhone    VARCHAR(50),
   @Email   VARCHAR(100),
   @Address VARCHAR(255),
   @Postcode    CHAR(6),
   @QQ  VARCHAR(50),
   @MSN VARCHAR(50),
   @ID   INT OUTPUT
)
AS

-- 确保数据的一致性，因而启用事务
BEGIN TRAN CustomerInsert
```

```
/* 插入用户*/
INSERT INTO Customers
(
   Name,
   Password,
   Phone,
   Fax ,
   MobilePhone,
   Email,
   Address,
   Postcode,
   QQ ,
   MSN
)
VALUES
(
   @Name,
   @Password,
   @Phone,
   @Fax,
   @MobilePhone,
   @Email,
   @Address,
   @Postcode,
   @QQ  ,
   @MSN
)

-- 返回当前所产生的CustomerID
SELECT
   @ID= @@Identity

COMMIT TRAN CustomerInsert
RETURN
```

3．删除：spCustomerDelete

```
ALTER PROCEDURE dbo.spCustomerDelete
(
   @ID  INT
)
AS
Delete From Customers Where  ID = @ID
```

4．修改：spCustomerUpate

```
ALTER PROCEDURE dbo.spCustomerUpate
(
    @ID        INT ,
    @Name       VARCHAR(50),
    @Password   VARCHAR(50),
    @Phone      VARCHAR(50),
    @Fax        VARCHAR(50),
    @MobilePhone     VARCHAR(50),
    @Email    VARCHAR(100),
    @Address VARCHAR(255),
    @Postcode    CHAR(6),
    @QQ  VARCHAR(50),
    @MSN VARCHAR(50)
)
AS

UPDATE Customers
SET
    Name = @Name,
    Password = @Password,
    Phone = @Phone,
    Fax = @Fax,
    MobilePhone = @MobilePhone,
    Email = @Email,
    Address = @Address,
    Postcode = @Postcode,
    QQ = @QQ,
    MSN = @MSN
WHERE
    ID= @ID
```

对于订单的插入和其他表的操作不同，因为在插入订单的过程中，要同时往订单详情表中插入信息，并且订单生成后要同时清空对应的购物车。该存储过程如下：

```
ALTER PROCEDURE dbo.spOrdersInsert
(
    @CartID    VARCHAR(50),
    @CustomerID INT,
    @OrderDate  SMALLDATETIME,
    @PayWay    VARCHAR(50),
```

```
    @CarryWay   VARCHAR(50),
    @Memo    VARCHAR(500),
    @ReceiveAdress VARCHAR(2000),
    @ID    INT OUTPUT
)
AS

-- 确保数据的一致性，因而启用事务
BEGIN TRAN OrdersInsert

/* 产生订单*/
INSERT INTO Orders
(
    CustomerID,
    OrderDate,
    PayWay,
    CarryWay,
    Memo,
    ReceiveAdress
)
VALUES
(
    @CustomerID,
    @OrderDate,
    @PayWay,
    @CarryWay,
    @Memo,
    @ReceiveAdress
)

-- 返回当前所产生的OrderID
SELECT
    @ID = @@Identity

/* 将当前用户的当前购物车COPY到当前所产生的订单详情表中*/
INSERT INTO OrderDetails
(
    OrderID,
    ProductID,
    Quantity,
    UnitCost
```

```
)

SELECT
    @ID,
    ShoppingCart.ProductID,
    Quantity,
    Products.UnitCost

FROM
    ShoppingCart
  INNER JOIN Products ON ShoppingCart.ProductID = Products.ProductID

WHERE
    CartID = @CartID

/* 完成后，清空当前购物车ID的所有明细*/
EXEC spShoppingCartEmpty @CartId

COMMIT TRAN OrdersInsert
```

这里就不再把存储过程源代码一一列出了，读者可以在本书配盘中找到这些源代码。

21.4.2　页面设计

系统界面风格基本一致，本着方便用户宗旨，在页面的设计上趋向于人性化，易于操作：

- 界面清晰明了、协调一致，同样功能用同样的图形。
- 有清楚的错误提示，误操作后，系统提供有针对性的提示。
- 使用用户的语言，而非技术的语言。
- 方便退出，随时转移界面，很容易从一个界面跳到另外一个界面。
- 让用户知道自己当前的位置，使其做出下一步行动的决定。

前台页面：

- default.aspx：商城首页
- CustomerReg.aspx：用户注册页面

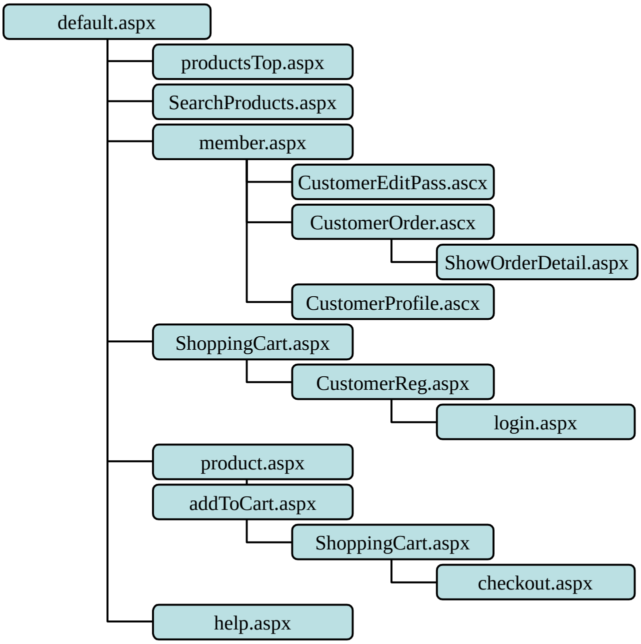
- login.aspx：用户登录页面
- member.aspx：会员页面
- CustomerEditPass.ascx：会员修改密码
- CustomerOrder.ascx：会员修改订单
- CustomerProfile.ascx：会员修改基本信息
- product.aspx：商品页面
- productsTop.aspx：特殊商品页面
- addToCart.aspx：选购商品
- ShoppingCart.aspx：购物清单
- SearchProducts.aspx：商品搜索
- checkout.aspx：订单确认页面
- ShowOrderDetail.aspx：显示订单详情
- help.aspx:帮助页面
- Errorpage.aspx：错误页面

后台页面：

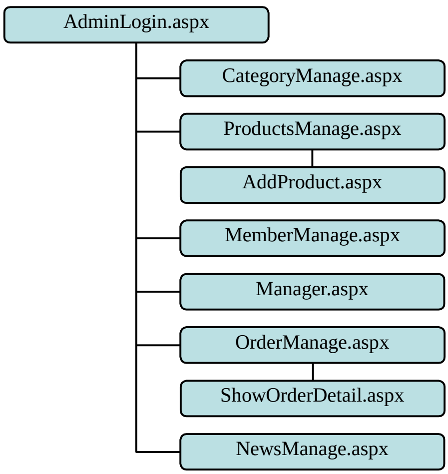
- AdminLogin.aspx：后台登录页面
- MemberManage.aspx：顾客信息管理页面
- Manager.aspx：管理员信息管理页面
- CategoryManage.aspx：商品种类管理页面
- ProductsManage.aspx：商品管理页面
- AddProduct.aspx：添加商品页面
- OrderManage.aspx：订单管理页面
- ShowOrderDetail.aspx：订单明细显示页面
- NewsManage.aspx：公告管理页面

21.4.3 页面关系图

1．前台页面



2．后台页面



21.4.4　界面设计

1．用户控件

在 ASP.NET 中用户控件就是一种组件。使用用户控件的好处是很明显的，它是面向对象和模块化思维的典型应用。一次编制可以重复使用，最大限度地体现了软件开发的优势。

在本系统中，我们主要设计以下用户控件：

- AdminHeader.ascx：用于后台管理页面最上端信息。



图 21-6　AdminHeader.ascx 用户控件

- Header.ascx：用于前台页面最上端信息。

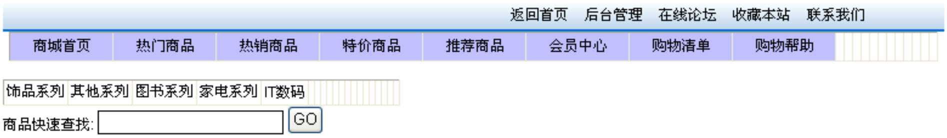


图 21-7　Header.ascx 用户控件

- NewestProduct.ascx：最新商品。

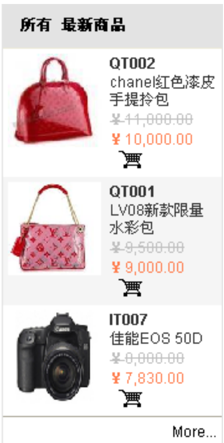


图 21-8　NewestProduct.ascx 用户控件

● TopProducts.ascx：热门商品。



图 21-9　TopProducts.ascx 用户控件

● Bulletin.ascx：公告显示。

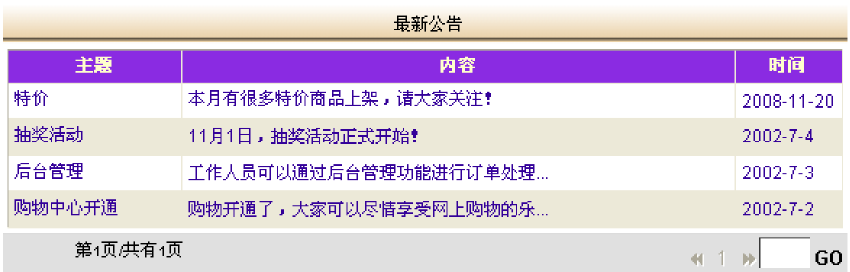


图 21-10　Bulletin.ascx 用户控件

● Commend.ascx：推荐商品



图 21-11　Commend.ascx 用户控件

- Footer.ascx：页面最底段信息。

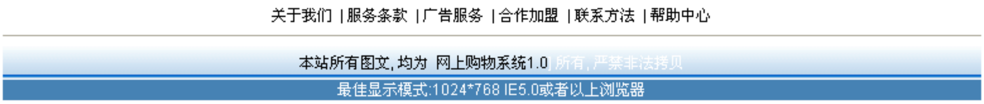


图 21-12　Footer.ascx 用户控件

2．前台页面

- default.aspx：商城首页。



图 21-13　商城首页

- CustomerReg.aspx：用户注册页面。

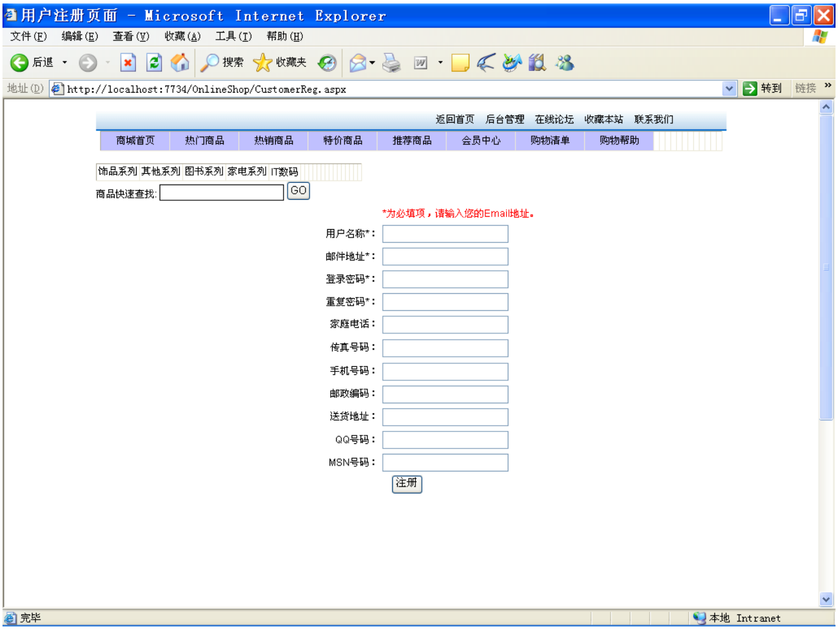


图 21-14 用户注册页面

- login.aspx：用户登录页面。

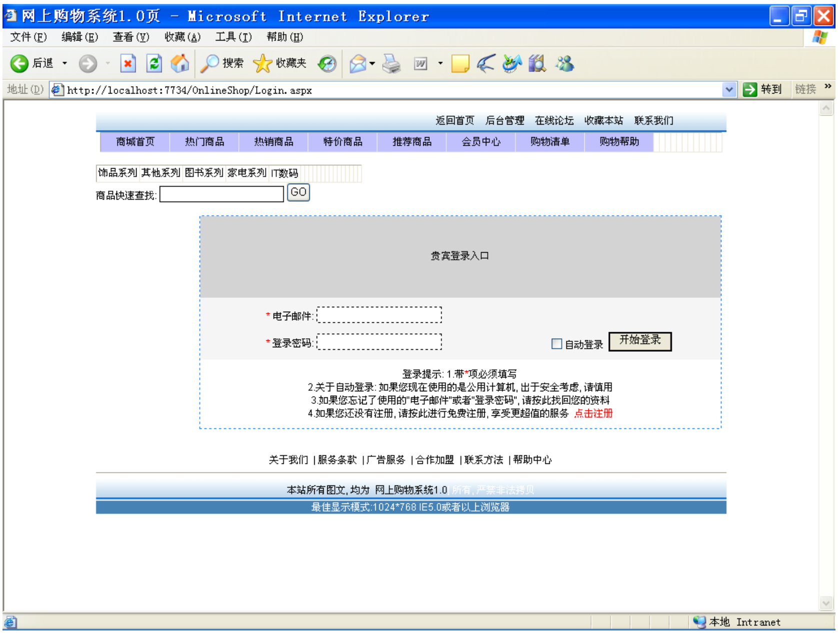


图 21-15 用户登录页面

- member.aspx—CustomerEditPass.ascx：会员修改密码。

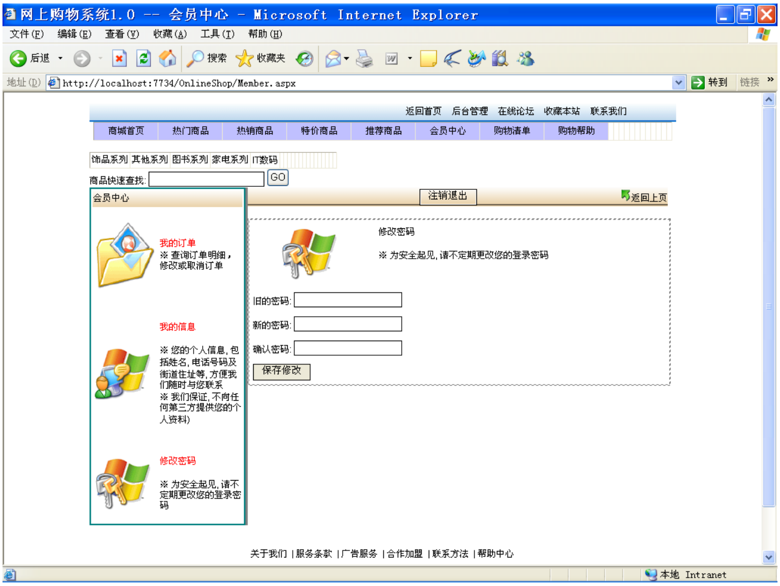


图 21-16 会员密码修改页面

- member.aspx—CustomerOrder.ascx：会员修改订单。



图 21-17 会员修改订单页面

- member.aspx—CustomerProfile.ascx：会员修改基本信息。

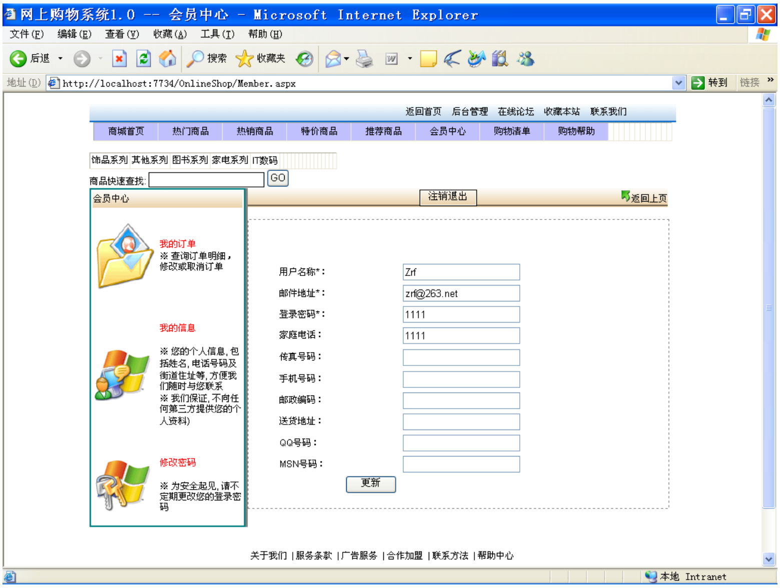


图 21-18　会员信息修改页面

- product.aspx：商品页面。



图 21-19　商品页面

- productsTop.aspx：特殊商品页面。



图 21-20　特殊商品页面

- ShoppingCart.aspx：购物清单页面。



图 21-21　购物清单页面

- SearchProducts.aspx：商品搜索页面。



图 21-22　商品搜索页面

- checkout.aspx：订单确认页面。



图 21-23　订单确认页面

- ShowOrderDetail.aspx：显示订单详情。

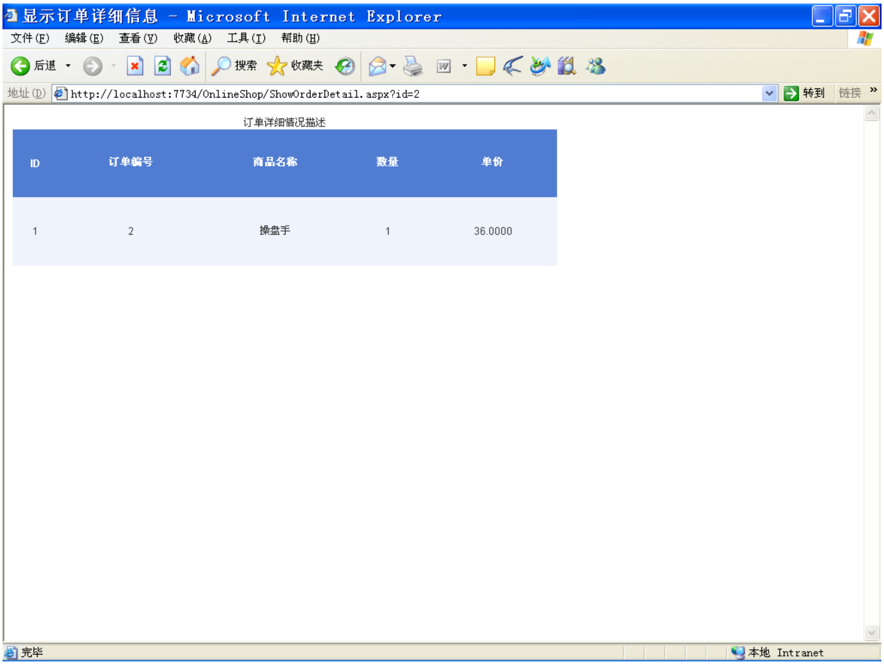


图 21-24　订单详情页面

3．后台页面

- AdminLogin.aspx：后台登录页面。

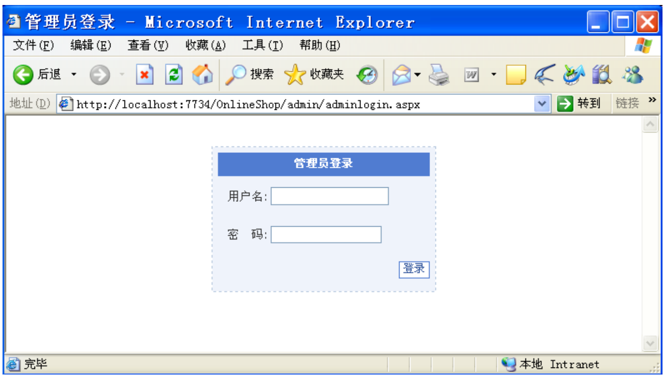


图 21-25　后台登录页面

- MemberManage.aspx：顾客信息管理页面。



图 21-26　顾客信息管理页面

- Manager.aspx：管理员信息管理页面。



图 21-27　管理员信息管理页面

- CategoryManage.aspx：商品种类管理页面。



图 21-28　商品种类管理页面

- ProductsManage.aspx：商品管理页面。



图 21-29　商品管理页面

- AddProduct.aspx：添加商品页面。



图 21-30　添加商品页面

- OrderManage.aspx：订单管理页面。



图 21-31　订单管理页面

- ShowOrderDetail.aspx：订单明细显示页面。



图 21-32　订单明细页面

- NewsManage.aspx：公告管理页面。



图 21-33　公告管理页面

21.5 编码和测试

21.5.1 公共模块设计

在编码阶段，首先要做的事就是要进行基础类和公共模块的设计（实际上，在详细设计最后阶段就要开始这个工作了）。

1. 数据库操作

一般情况下，我们都把一些数据库操作代码封装到公共类中，同时也把一些共同的操作放入公共类。

```
    public class Pub
    {
       // 数据库公用变量
       public SqlConnection con;
       public SqlCommand com;
       public SqlDataAdapter da;
       public DataSet ds;
       public DataTable dt;
       public DataRow dr;

       // 非数据库公用变量
       public string LoginName;      // 登录用户
       public   string   SiteName   =   ConfigurationSettings.AppSettings
["SiteName"];                        // 站点名称
       public string SiteNav;        // 站点导航
       public string sTitle = ConfigurationSettings.AppSettings["SiteName"];
                                     // 站点标题

       // 页面导航
       public int PageSize = 5;      // 每页显示记录数
       public int RecordCount, PageCount, CurrentPage;

       public Pub()
       {
          //设置应用程序目录
          string p = AppDomain.CurrentDomain.BaseDirectory;
          string DataDirectory = p+ "database\\";
          AppDomain.CurrentDomain.SetData("DataDirectory",
DataDirectory);
      }
```

```
        public void connOpen()
        {
            con = new SqlConnection(ConfigurationSettings.AppSettings["conn"]);
            con.Open();
        }

        /// <summary>
        /// 连接关闭(之前清除 ds 中的数据)
        /// </summary>
        public void connClose()
        {
            if (ds != null)
            {
                ds.Clear();
            }
            if (con != null)
            {
                con.Close();
            }
        }

        /// <summary>
        /// 建立 DataSet 对象，用记录填充或构架（如果必要）DataSet 对象,DataSet 即是
数据在内存的缓存
        /// </summary>
        /// <param name="sSQL">打开表 Sql 语句</param>
        public void Fill(string sSQL)
        {
            connOpen();
            da = new SqlDataAdapter(sSQL, con);
            ds = new DataSet();
            da.Fill(ds);
        }

        /// <summary>
        /// 引用存储过程构造 DataSet
        /// </summary>
        /// <param name="sp">要引用的存储过程</param>
        public void FillDataSet(string sp)
        {
            connOpen();
            da = new SqlDataAdapter(sp, con);
            ds = new DataSet();
            da.Fill(ds);
        }
```

```
        /// <summary>
        /// 建立DataSet对象,用记录填充或构架(如果必要)DataSet对象,DataSet即是数据在内存的缓存
        /// </summary>
        /// <param name="TableName">用于表映射的源表的名称</param>
        /// <param name="sql">打开表 Sql 语句</param>
        public void Fill(string TableName, string sql)
        {
            connOpen();
            da = new SqlDataAdapter(sql, con);
            ds = new DataSet();
            da.Fill(ds, TableName);
        }

        /// <summary>
        /// 给 DataSet 对象增加映射表
        /// </summary>
        /// <param name="sSQL">SQL 语句</param>
        public void FillAdd(string sSQL)
        {
            da = new SqlDataAdapter(sSQL, con);
            da.Fill(ds);
        }

        /// <summary>
        /// 给 DataSet 对象增加映射表
        /// </summary>
        /// <param name="sSQL">SQL 语句</param>
        public void FillAdd(string TableName, string sSQL)
        {
            da = new SqlDataAdapter(sSQL, con);
            da.Fill(ds, TableName);
        }

        /// <summary>
        /// 获取最大值
        /// </summary>
        /// <param name="fID"></param>
        /// <param name="TableName"></param>
        /// <returns></returns>
        public string GetMaxId(string fID, string TableName)
        {
            string sKey;
            string sSQL = "SELECT TOP 1 " + fID + " FROM " + TableName + " ORDER
```

```
BY " + fID + " DESC";
            if (GetRowCount(sSQL) == 0)
            {
                sKey = "1";
            }
            else
            {
                GetRowRecord(sSQL);
                sKey = (int.Parse(dr[fID].ToString()) + 1).ToString(); // 获
得数据库表 key 值
            }
            return sKey;
        }

        /// <summary>
        /// 获得包含在 DataSet 对象的映射表集合中的 index 为 0 的映射表
        /// </summary>
        /// <param name="sSQL">SQL 语句</param>
        public void GetTable(string sSQL)
        {
            Fill(sSQL);
            dt = ds.Tables[0];
        }

        /// <summary>
        /// 获得符合该 Sql 语句的表记录数
        /// </summary>
        /// <param name="sSQL">SQL 语句</param>
        /// <returns>返回表记录条数</returns>
        public int GetRowCount(string sSQL)
        {
            Fill(sSQL);
            try
            {
                int Count = ds.Tables[0].Rows.Count;
                connClose();
                return Count;
            }
            catch
            {
                connClose();
                return 0;
            }
        }
```

```
        /// <summary>
        /// 通过传 Sql 语句关键 key 值获得表中一行的数据
        /// </summary>
        /// <param name="sSQL">带关键 Key 值参数的 SQL 语句</param>
        public void GetRowRecord(string sSQL)
        {
            Fill(sSQL);
            dr = ds.Tables[0].Rows[0];
            connClose();
        }

        /// <summary>
        /// 执行 Transact-SQL 语句,对数据库记录做插入,修改,删除等操作
        /// </summary>
        /// <param name="sSQL">Transact-SQL 语句</param>
        public void ExeSql(string sSQL)
        {
            connOpen();
            com = new SqlCommand(sSQL, con);
            com.ExecuteNonQuery();
            com.Dispose();
    }
    /// <summary>
        /// 绑定 DataGrid 控件并显示数据
        /// </summary>
        /// <param name="sSQL">Sql 语句</param>
        /// <param name="DataGrid1">DataGrid 控件 id 值</param>
        public void BindDataGrid(string sSQL, DataGrid dg)
        {
            Fill(sSQL);
            dg.DataSource = ds.Tables[0].DefaultView;
            dg.DataBind();
        }

        /// <summary>
        /// 绑定 DataList 控件并显示数据
        /// </summary>
        /// <param name="sSQL">Sql 语句</param>
        /// <param name="DataList1">DataList 控件 ID 值</param>
        public void BindDataList(string sSQL, DataList dl)
        {
            Fill(sSQL);
            dl.DataSource = ds.Tables[0].DefaultView;
            dl.DataBind();
        }
```

```
        /// <summary>
        /// 绑定 Repeater 控件并显示数
        /// </summary>
        /// <param name="sSQL">Sql 语句</param>
        /// <param name="Repeater1">Repeater 控件 ID 值</param>
        public void BindRepeater(string sSQL, Repeater rt)
        {
            Fill(sSQL);
            rt.DataSource = ds.Tables[0].DefaultView;
            rt.DataBind();
        }
    }
```

2. 存储过程调用函数

对于在前面设计好的存储过程，我们在公共模块中添加对应的方法。例如，对于 Customers 的几个存储过程，我们在 Pub.cs 公共模块中设计操作函数如下：

```
    /// <summary>
        /// 用户登录
        /// </summary>
        /// <param name="sEmail">登录时的电子邮件</param>
        /// <param name="sPassword">相应的密码</param>
        /// <returns></returns>
        public String spCustomersLogin(string sEmail, string sPassword)
        {
            connOpen();
            com = new SqlCommand("spCustomersLogin", con);
            com.CommandType = CommandType.StoredProcedure;

            SqlParameter parEmail = new SqlParameter("@Email", SqlDbType.
VarChar, 100);
            SqlParameter parPass = new SqlParameter("@Password", SqlDbType.
VarChar, 50);
            SqlParameter parID = new SqlParameter("@ID", SqlDbType.Int, 4);
            com.Parameters.Add(parEmail);
            com.Parameters.Add(parPass);
            com.Parameters.Add(parID);
            parEmail.Value = sEmail;
            parPass.Value = sPassword;
            parID.Direction = ParameterDirection.Output;

            com.ExecuteNonQuery();
            int ID = (int)(parID.Value);
```

```
            if (ID == -1)
                return null;
            else
                return ID.ToString();
        }

        /// <summary>
        /// 插入新用户
        /// </summary>
        /// <returns></returns>
        public string spCustomerInsert(string sName, string sPassword,string
sPhone,string sFax,
            string   sMobilePhone,string   sEmail,string   sAddress,string
sPostcode,
            string sQQ,string sMSN)
        {
            connOpen();
            com = new SqlCommand("spCustomerInsert", con);
            com.CommandType = CommandType.StoredProcedure;

            SqlParameter parName = new SqlParameter("@Name", SqlDbType.VarChar,
50);
            SqlParameter parPassword = new SqlParameter("@Password", SqlDbType.
VarChar, 50);
            SqlParameter  parPhone  =  new  SqlParameter("@Phone",  SqlDbType.
VarChar, 50);
            SqlParameter parFax = new SqlParameter("@Fax", SqlDbType.VarChar, 50);
            SqlParameter  parMobilePhone  =  new  SqlParameter("@MobilePhone",
SqlDbType.VarChar, 50);
            SqlParameter  parEmail  =  new  SqlParameter("@Email",  SqlDbType.
VarChar, 100);
            SqlParameter parAddress = new SqlParameter("@Address", SqlDbType.
VarChar, 255);
            SqlParameter parPostcode = new SqlParameter("@Postcode", SqlDbType.
Char,6);
            SqlParameter parQQ = new SqlParameter("@QQ", SqlDbType.VarChar, 50);
            SqlParameter parMSN = new SqlParameter("@MSN", SqlDbType.VarChar, 50);
            SqlParameter parID = new SqlParameter("@ID", SqlDbType.Int, 4);
            com.Parameters.Add(parName);
            com.Parameters.Add(parPassword);
            com.Parameters.Add(parPhone);
            com.Parameters.Add(parFax);
            com.Parameters.Add(parMobilePhone);
            com.Parameters.Add(parEmail);
            com.Parameters.Add(parAddress);
```

```
            com.Parameters.Add(parPostcode);
            com.Parameters.Add(parQQ);
            com.Parameters.Add(parMSN);
            com.Parameters.Add(parID);
            parName.Value = sName;
            parPassword.Value = sPassword;
            parPhone.Value = sPhone;
            parFax.Value = sFax;
            parMobilePhone.Value = sMobilePhone;
            parEmail.Value = sEmail;
            parAddress.Value = sAddress;
            parPostcode.Value = sPostcode;
            parQQ.Value = sQQ;
            parMSN.Value = sMSN;
            parID.Direction = ParameterDirection.Output;

            com.ExecuteNonQuery();
            int ID = (int)(parID.Value);
            if (ID == 0)
               return null;
            else
               return ID.ToString();
        }
        /// <summary>
        /// 删除指定顾客记录
        /// </summary>
        /// <returns></returns>
        public void spCustomerDelete(int nID)
        {
            connOpen();
            com = new SqlCommand("spCustomerDelete", con);
            com.CommandType = CommandType.StoredProcedure;

            SqlParameter parID = new SqlParameter("@ID", SqlDbType.Int);
            com.Parameters.Add(parID);
            parID.Value = nID;
            com.ExecuteNonQuery();
        }
        /// <summary>
        /// 更新用户信息
        /// </summary>
        /// <returns></returns>
        public void spCustomerUpdate(int nID, string sName, string sPassword,
string sPhone,string sFax,
            string   sMobilePhone,string   sEmail,string   sAddress,string
```

```
sPostcode,string sQQ,string sMSN)
        {
            connOpen();
            com = new SqlCommand("spCustomerUpate", con);
            com.CommandType = CommandType.StoredProcedure;

            SqlParameter parName = new SqlParameter("@Name", SqlDbType.VarChar,
50);
            SqlParameter parPassword = new SqlParameter("@Password", SqlDbType.
VarChar, 50);
            SqlParameter parPhone = new SqlParameter("@Phone", SqlDbType.
VarChar, 50);
            SqlParameter parFax = new SqlParameter("@Fax", SqlDbType.VarChar, 50);
            SqlParameter parMobilePhone = new SqlParameter("@MobilePhone",
SqlDbType.VarChar, 50);
            SqlParameter parEmail = new SqlParameter("@Email", SqlDbType.
VarChar, 100);
            SqlParameter parAddress = new SqlParameter("@Address", SqlDbType.
VarChar, 255);
            SqlParameter parPostcode = new SqlParameter("@Postcode", SqlDbType.
Char, 6);
            SqlParameter parQQ = new SqlParameter("@QQ", SqlDbType.VarChar, 50);
            SqlParameter parMSN = new SqlParameter("@MSN", SqlDbType.VarChar, 50);
            SqlParameter parID = new SqlParameter("@ID", SqlDbType.Int, 4);
            com.Parameters.Add(parName);
            com.Parameters.Add(parPassword);
            com.Parameters.Add(parPhone);
            com.Parameters.Add(parFax);
            com.Parameters.Add(parMobilePhone);
            com.Parameters.Add(parEmail);
            com.Parameters.Add(parAddress);
            com.Parameters.Add(parPostcode);
            com.Parameters.Add(parQQ);
            com.Parameters.Add(parMSN);
            com.Parameters.Add(parID);
            parName.Value = sName;
            parPassword.Value = sPassword;
            parPhone.Value = sPhone;
            parFax.Value = sFax;
            parMobilePhone.Value = sMobilePhone;
            parEmail.Value = sEmail;
            parAddress.Value = sAddress;
            parPostcode.Value = sPostcode;
            parQQ.Value = sQQ;
            parMSN.Value = sMSN;
```

```
            parID.Value = nID;

            com.ExecuteNonQuery();

        }

         /// <summary>
        /// 更新用户密码
        /// </summary>
        /// <param name="sEmail">用户 ID</param>
        /// <param name="sPassword">相应的密码</param>
        /// <param name="sPasswordNew">新的密码</param>
        /// <returns></returns>
        public String spCustomersUpdatePassword(int nID, string sPassword,
string sPasswordNew)
        {
            connOpen();
            com = new SqlCommand("spCustomersUpdatePassword", con);
            com.CommandType = CommandType.StoredProcedure;

            SqlParameter parID  = new SqlParameter("@ID", SqlDbType.Int);
            SqlParameter  parPassOld  =  new  SqlParameter("@PasswordOld",
SqlDbType.VarChar, 50);
            SqlParameter  parPassNew  =  new  SqlParameter("@PasswordNew",
SqlDbType.VarChar, 50);
            SqlParameter parResult = new SqlParameter("@Result", SqlDbType.
VarChar, 100);
            com.Parameters.Add(parID);
            com.Parameters.Add(parPassOld);
            com.Parameters.Add(parPassNew);
            com.Parameters.Add(parResult);
            parID.Value = nID;
            parPassOld.Value = sPassword;
            parPassNew.Value = sPasswordNew;
            parResult.Direction = ParameterDirection.Output;

            com.ExecuteNonQuery();
            return parResult.Value.ToString();
        }
```

这样的好处是：我们把所有有关数据库的操作都封装在了一个类当中，便于修改，也便于整个项目组内人员的协调。

其他存储过程的调用代码我们不再列出，读者可以查看本实例的源代码中 App_Code 目录下 pub.cs 文件。

21.5.2 源代码控制

源代码控制工具同样适用微软的产品 Visual SourceSafe，它可以很好地集成在 VisualStudio.NET 中。在工具篇中，我们已经简单介绍了 VSS 的使用。

21.5.3 单元测试工具

由于我们使用.NET 开发平台，因此我们使用 NUnit 作为单元测试工具。在工具篇中，我们已经简单介绍了 NUnit 的使用。

21.5.4 系统源代码

在本书的配套光盘中“Demo\project”目录下可以找到源代码。

21.6 制作安装盘

我们使用 Visual Studio.NET 本身来制作程序安装盘。在工具篇中，我们已经介绍了使用 VS.NET 制作安装盘的过程，在此不再赘述，大家可以参考前面的内容完成。

制作好的安装程序可以在本书附带光盘中找到。

21.7 系统源文件

在本书配盘 DEMO\Project 目录下可以找到本系统源文件。

目录说明如下：

- \Rose：UML 图源文件。
- \SQLExpress：SQL Server2005 Express 和 SQL Server Management Studio Express 下载地址。
- \Database：数据库文件。
- \OnlineShop：程序源文件、软件加密源文件。
- \ WebSetup：安装盘制作源文件、安装盘。

第4篇 模板与规范篇

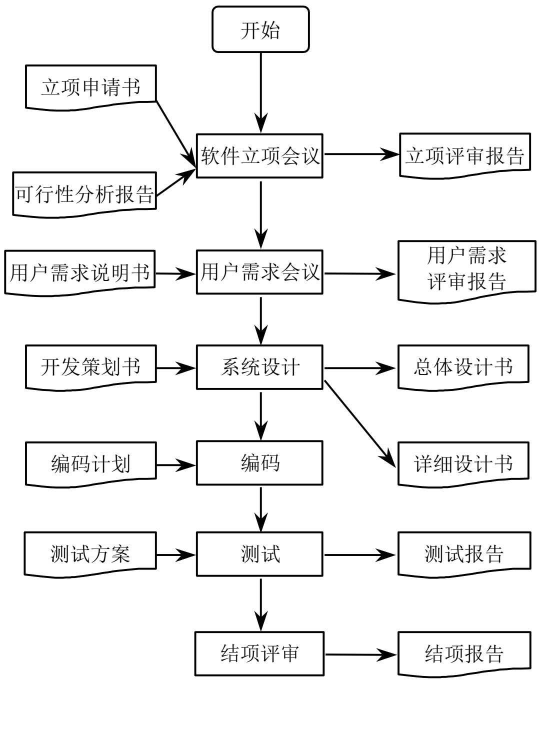
本篇导读

在本篇中，我们提供了一套简化后的质量模板文件，比较适合中小型软件企业的应用。还提供了团队开发所需要的各种开发规范，包括项目组工作守则、编码规范、源代码控制规范、界面设计规范、数据库使用规范、安装盘制作规范等，会极大地方便团队开发。为了便于大家使用，我们在本书配带的光盘中提供了这些模板和规范的电子版。

第 22 章　软件开发模板

本章导读：正如知识篇中所述，质量标准和效率之间永远存在着矛盾。实际应用中主要要考虑它们之间的平衡：既不能完全不顾质量标准，任意而为；又不能每个阶段全部严格按照标准而不考虑开发效率。

在本章中，我们提供了一套简化后的质量模板文件，比较适合中小型软件企业的应用。为了便于大家使用，我们在本书配带的光盘中提供了这些规范的电子版。这套模板的应用情况如下。



22.1 软件立项申请书

<table>
<tr><td colspan="2">软件名称</td><td colspan="8"></td></tr>
<tr><td colspan="10">参加人员基本情况</td></tr>
<tr><td colspan="2">立项部门</td><td colspan="2"></td><td colspan="2">发起人</td><td colspan="4"></td></tr>
<tr><td>姓名</td><td></td><td>性别</td><td></td><td>年龄</td><td></td><td>民族</td><td></td><td>职务</td><td></td></tr>
<tr><td colspan="2">职　称</td><td colspan="2"></td><td>专业</td><td colspan="5"></td></tr>
<tr><td colspan="2">工作单位</td><td colspan="2"></td><td>电话</td><td colspan="5"></td></tr>
<tr><td colspan="2">通信地址</td><td colspan="2"></td><td>邮编</td><td colspan="5"></td></tr>
<tr><td colspan="10">主　要　参　加　人　员</td></tr>
</table>

姓名	年龄	职务职称	专　业	工作单位	工作任务分工

续表

软件立项的起因、背景、意义		
预期达到的目的		
研发本软件的基本思路及关键点		
技术上的可行性分析（可以提交《技术分析报告》。）		
市场可行性分析（可以提交《市场分析报告》）		
开发进度安排（开发计划时间进度表，阶段性成果承诺）		
预计投入资金	总额	
	使用安排	
	备　注	
验收标准（包括产品质量、性能、进度、投入等。可单独提交《验收标准》）		
风险分析		
发起人签名：　　　　　　　　年　　月　　日		

22.2 可行性分析报告

1．市场可行性分析

（1）市场定位

软件卖给谁、卖到哪里，软件综合水平。

（2）总体市场情况

市场发展历史与趋势，市场总份额、市场占有率情况，本产品处于市场的什么发展阶段。

（3）竞争对手分析

同类软件比较、厂家比较、价格分析、市场占有率分析。

（4）用户分析

消费群体特征、消费方式、用户现有技术情况以及影响市场的因素分析。

（5）市场策略

预计采取的市场策略。

2．政策可行性分析

（1）政策“支持”

分析是否有有利政策，政策适用条件分析。

（2）政策“限制”

分析是否有政策限制，限制的具体方式是什么。

3．技术和时间分析

（1）公司相关部门的技术储备和技术能力等。

（2）技术方案及评估

- 关键技术问题
- 各方案对比分析

- 确定本项目拟采用的技术方案：（硬件平台、软件平台、开发工具、辅助工具、对开发人员的基本技术要求等）
- 备选方案
- 技术可行性评价：
  - 在限制条件下，功能目的是否达到
  - 利用现有技术，功能目的是否达到
  - 对开发人员数量和质量的要求，并说明能否满足
  - 在规定的期限内，开发能否完成

（3）问题及风险

列出所采用方案的主要风险及规避措施。

4．成本－收益分析

（1）总投入：

包括人员投入、资金投入、软硬件投入、其他投入等。

（2）估计收益：

软件能卖多少年；推广第一年投入产出分析；推广第三年投入产出分析。

5．SWOT 分析

（1）我们的强项是什么？如何利用这些强项？

（2）我们的弱项是什么？如何减少这些弱项的影响？

（3）市场中什么样的机会？如何把握这些机会？

（4）有哪些威胁？如何有效地应付这些威胁？

22.3 立项评审报告

1．立项软件简介

包括软件名称、功能、用户对象、申请立项的部门、发起人等。

2．软件评审

市场竞争力评价

预算和收入评价

可行性分析

不确定因素及解决办法

验收标准

3．评审活动

会议举行日期、参与人员

会议流程简介

参与人员意见

评审结论

4．负责人签名

包括项目负责人、公司高层意见

22.4 用户需求说明书

1．项目描述

（1）描述本项目的适用场合及处理业务。

（2）项目名称：本项目的名称，包括项目的全名、简称、代号、版本号。

（3）名词定义：对重要的或是具有特殊意义的名词进行定义。

2．调研情况介绍

描述主要的调研活动及对象。

3．用户情况描述

（1）用户业务描述：描述本项目的用户使用本项目处理的业务。

（2）用户情况：介绍本项目的用户的情况，包括：

- 用户的工作流程；
- 用户的相关部门及职责；
- 用户的技术水平。

（3）用户原有系统的情况：介绍用户现在使用的系统的主要情况，包括主要的不足。

4．任务概述

（1）目标

阐明本项目所需达到的目的。

（2）运行环境

- 硬件环境

详细列出本软件运行时所必需的最低硬件配置、推荐硬件配置（如主机、显示器、外部设备等）以及其他特殊设备。

- 软件环境

如操作系统、网络软件、数据库系统以及其他特殊软件要求。

（3）条件与限制

说明本软件在实现时所必须满足的条件和所受的限制，以及相应的原因。必须满足的条件包括输入数据的范围以及格式，所受的限制包括软件环境、硬件环境等方面的内容。

5．功能需求

（1）功能划分

从用户的角度将产品按功能划分成不同的部分，但应注意此处划分成的部分并不对应于最终程序实现时的不同功能模块。

（2）功能描述

细化由功能划分所生成的各部分的内容，应包括下列内容：

1）必须完成的功能以及对此功能的详细描述：

按功能类型分类，逐条列出本软件所能完成的各项功能以及对此功能的详细描述。

2）不支持的功能以及相应的原因：

- 列出本软件所不支持的各项功能以及相应的原因。
- 此部分内容务必详细准确、无二义性，以作为将来验收和测试的标准。

6．数据描述

（1）输入/输出数据

说明输入/输出数据的类型及格式。

（2）数据流图

从数据传递和加工的角度描述的数据流图，此数据流图不包含任何有关实现的内容，只是从最上层对有关内容加以描述。数据流图的表述形式参见软件工程中的有关规定。

（3）数据库描述（可选）

根据系统的总目标和范围，定义数据库的逻辑特性及物理特性。

（4）数据词典

对于数据流图中出现所有被命名的图形元素在数据词典中作为一个词条加以定义，使得每一个图形元素的名字都有一个确切的解释。

7．性能需求

（1）数据精确度

根据实际情况，确定软件最终输出数据（包括传输中）的数据精确度。

（2）时间特性

说明开发的软件在响应时间、更新处理时间、数据转换与传输时间、运行时间等方面所需达到的时间特性。

（3）适应性

（4）复用性

说明本项目是否可以复用已有软件、是否可为其他产品复用。

（5）灵活性

说明在运行环境、与其他软件的接口，以及开发计划等发生变化时，应具有的适应能力。

8．运行需求

（1）用户界面

说明本软件的人机界面风格。

（2）硬件接口

说明该软件与硬件之间各接口的逻辑特点及运行该软件的硬件设备特征。

（3）软件接口

说明该软件与其他软件之间接口，对于每个需要的软件，应提供

- 接口名称；
- 规格说明；
- 版本号。

（4）故障处理

说明本软件在健壮性方面所需达到的目标。

9．不确定的问题

说明目前尚未确定的问题及处理的计划。

10．风险分析

说明本项目面临的主要风险，包括时间、技术复杂度、人力资源等。

11．其他需求

如可使用性、安全保密、可维护性、可移植性等。

特殊要求：

由用户提出的，或是本公司要求的特殊要求、特殊的情况等。

12．参考资料

填写要求：列出参考引用的资料的名称，并附原文。

13．编写人员及编写日期

列出参与编写的人员的名字，并标明负责人。

22.5 需求分析评审报告

软件名称	用户名称
评审委员会： 负责人：	
评审日期：　　　　年　月　日	
评审活动记录：	
不确定事项及处理办法：	
评审结论：	
负责人签名： 日期：　　　年　月　日	
产品经理/项目经理意见： 签名：　　　　　　　　日期：　　　年　月　日	
备注：	

22.6 总体设计书

1．软件概述

对整个软件的简介，包括名称和功能。

2．软件开发环境

介绍软件的开发平台、工具以及网络环境配置等。

3．系统的业务流程和数据流程

介绍软件的业务流程和数据流程、数据汇总图等。

4．系统总体结构及模块划分

介绍整个软件系统的总体结构、模块划分、每个模块的功能。

5．系统画面迁移图

介绍整个软件系统各个模块的功能调用图。

6．系统与其他系统的接口

7．系统各模块之间的接口

8．系统各模块功能及详细描述，实现方法

9．实现模块功能的关键算法和关键技术

包括关键技术的提出、关键技术的一般说明、关键技术的实现方案等。

10．对详细设计的一些具体要求

如关键功能的性能指标下限，设计用户界面时的主要规范，对某些功能或模块在实现方法和开发工具上的限制和制定等。

11．人员配置及进度计划

介绍整个总体设计阶段的人员配置、分工、进度计划以及验收标准。

22.7 开发策划书

1．软件描述

描述软件的背景、开发目标。

2．项目人员职责

确定开发组中的开发经理、测试经理、开发人员。

3．开发活动计划

（1）项目开发资源描述

描述项目开发中需要的资源，如人员配备、职责与权限、设备与工具。

（2）开发阶段的划分和进度安排

明确项目开发主要的阶段，包括阶段的划分、阶段的输入、输出及验证准则。规定组成项目开发的各任务实施的时间表、相互的进展顺序以及不同任务和阶段对资源的要求。

（3）开发方法和工具

a．描述本项目开发中的使用的主要的工具和技术方法。

b．描述本项目中使用到的规则、惯例和约定，特别是本项目中使用的由本项目组或所属产品部制定的内部规定及指导书等。

（4）各部门间的接口与协调及其他

描述相关部门如公司、测试人员等应提供的协助。

4．测试计划

（1）各阶段测试目标及要求

描述本项目需进行的单元测试、测试、稳定期测试目标和要求。

（2）各测试阶段资源要求及时间安排

对项目各阶段测试所需要的测试人员的数量与相应培训要求、测试用的时间进行预计安排。

（3）测试记录的具体要求

规定本项目各类测试活动的测试记录的填写要求。

5．确认签名

开发经理签名：　　　　　　日期：

测试经理签名：　　　　　　日期：

22.8 详细设计书

描述软件详细的程序结构、处理流程，至少应包括以下内容：

1. 程序流程结构

描述软件总体、各个模块的程序结构、处理流程。

2. 接口

本模块与外部系统接口的要求及数据格式。

3. 数据格式

系统中的全局的或是重要的局部数据的定义及格式要求。

4. 模块及算法

描述主要的程序模块以及关键算法的实现方法。

5. 关键部件

系统中核心的或者重要的数据结构、组件或算法的说明。

6. 模块界面设计

7. 数据库系统设计

包括数据访问频度和流量、数据库选型、异构数据库的连接与数据传递方式、数据共享方式设计、数据安全性及保密设计、数据字典设计。

8. 技术难点

对截至目前遇到并已经解决的技术难题加以总结。

9. 人员配置及进度计划

介绍整个详细设计阶段的人员配置、分工、进度计划，以及验收标准。

22.9 编码计划

1. 编码人员及负责人

包括项目负责人、项目成员。

2．编码工作分工

介绍编码阶段项目组中各个成员工作分工。

3．工作进度安排

介绍整个编码阶段的进度安排，包括开始、验收时间，各个里程碑时间。

4．采用的工具

介绍项目组使用的开发工具、编程语言、测试工具、源代码管理工具等。

5．编码具体要求

介绍编码要求，包括编程规范、界面设计规范、源代码控制工具使用规范等。

6．输出要求

介绍编码阶段最后的输出情况。

7．计划制定人姓名及制定日期

20.10 测试方案

1．项目描述

本项目的名称，包括项目的全名、简称、代号、版本号。描述本项目的适用场合及处理业务。

2．测试需求的分析

对所测试软件的功能特点进行简要的描述，并针对其特点进行测试需求的分析，进一步得出对此软件测试的重点和要求，进行用户测试的应分析出用户测试的重点和不足。此部分分析深度和广度受测试项目的大小和重要程度的影响。

3．测试设计

根据已经确定的测试需求和测试重点详细的将需要测试的内容、测试方法、测试用例、测试程度的要求进行设计和描述，从而达到控制整个测试活动内容的目的。

4．测试计划

根据测试需求、测试设备环境（软硬），人力资源配备、测试时间等，进行分析和规划，做出整个测试活动的人员工作、进度安排，建议用 Project 作此部分工作。

5．用户测试规划

对于需要进行用户测试的测试项目，应该进行用户测试规划，此部分内容包括测试用户的选取条件，测试用户的情况（包括测试用户的特点和不足），负责用户测试的人员和采取的联系方法，用户测试的时间安排，用户测试版本的更新安排等内容。用户测试不是必需的，但针对重要项目或重要的版本应该注意选取合适的用户进行有效的用户测试。

6．编写人员及编写日期

列出参与编写的人员的名字，并标明负责人。

20.11 测试报告

软件名称：		
软件编号：	开发组名称：	开发经理:
版本号：	构造号：	软件语言：
开发工作量：(人日)		
本次版本升级修改过的代码行数（含增加、删除、修改）：（千行）		
软件总的代码行数：（千行）		
软件简介：		
软件建议配置：		
加密	产品加密方案：	
	加密方式：	
	加密测试结果：	
软件组成模块清单：		
软件组成文件清单：		
软件组成介质清单：		
测试时间段：		
测试人员：		
测试工作量：(人日)		
测试环境：		

续表

<table>
<tr><td colspan="6">测试活动简述：</td></tr>
<tr><td rowspan="3">测试错误分类</td><td></td><td>死机</td><td>致命</td><td>严重</td><td>一般</td></tr>
<tr><td>发现</td><td></td><td></td><td></td><td></td></tr>
<tr><td>改正</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2" rowspan="2">软件继承性</td><td colspan="2">继承软件名称：</td><td colspan="2">版本号：</td></tr>
<tr><td colspan="4">对继承软件的改进/改变处：
Bug 的修改：
功能的改进：
新功能的增加：
为配合相关的插件或软件做的改动：
针对某些功能模块做的优化：
其他：</td></tr>
<tr><td colspan="6">软件遗留问题（发生原因、对功能性能有何影响、解决措施）：</td></tr>
<tr><td colspan="6">测试组测试结论：</td></tr>
<tr><td rowspan="2">软件度量</td><td colspan="5">测试期发现的 bug 共　　个；建议共　　个，采纳　　个
Bug 密度（系统测试的 Bug 数/修改的代码行数）：　　个/千行
Bug 密度（系统测试的 Bug 数/总的代码行数）：　　个/千行</td></tr>
<tr><td colspan="5">待发行产品遗留 Bug 共　　个；严重及严重以上遗留 Bug 共　　个
Bug 密度(遗留 Bug 数/修改的代码行数)：　　个/万行
Bug 密度（遗留 Bug 数/总的代码行数）：　　个/万行
Bug 密度（严重及严重以上遗留 Bug 数/总的代码行数）：　　个/万行</td></tr>
<tr><td rowspan="2">测试综合报告评审</td><td colspan="5">评审方式、人员、时间：</td></tr>
<tr><td colspan="5">评审结论：</td></tr>
<tr><td>测试经理</td><td colspan="2">签名:</td><td colspan="3">日期:　　年　　月　　日</td></tr>
<tr><td rowspan="2">开发经理</td><td colspan="5">意见：</td></tr>
<tr><td colspan="2">签名：</td><td colspan="3">日期：　　年　　月　　日</td></tr>
<tr><td rowspan="2">产品经理</td><td colspan="5">意见：</td></tr>
<tr><td colspan="2">签名：</td><td colspan="3">日期：　　年　　月　　日</td></tr>
<tr><td colspan="6">备注：</td></tr>
</table>

22.12 软件开发结项报告

1．软件描述

简单介绍软件名称，中英文简称。

2．综合评估

对整个软件的开发过程以及结果做整体评估。

3．软件开发实施中计划的执行情况

描述软件开发实施阶段制定的项目实施策划开发计划、测试计划、配置管理计划和质量控制计划执行情况，并说明有无重大的修正和时间上的延迟。

4．软件质量目标完成情况

总结产品质量控制计划中拟定的质量目标的完成情况，若未完成，分析原因。

5．投入产出分析

整个软件开发过程中总的投入和产生的效益分析。

6．总结开发活动中的经验与教训

总结本软件开发过程中取得的成功经验，以及需要今后改进的地方。如质量控制、进度控制，与其他部门或用户的配合方面等。

7．维护工作的安排

明确维护负责人，维护的范围、职责、维护方式，及对维护记录的要求。

对于升级软件维护工作的安排要考虑产品开发的延续性。新产品结项时维护计划应单独形成文件，作为结项报告的附件。

8．人员以及日期

包括编写人及日期、审批人及日期。

第 23 章　项目组规范

本章导读：在本章中，我们提供了团队开发所需要的各种开发规范，包括项目组工作守则、编码规范、源代码控制规范、界面设计规范、数据库使用规范、安装盘制作规范等，会极大地方便团队开发。为了方便大家使用，我们在本书配套光盘中提供了这些规范的电子版。

23.1 项目组工作守则

为了促进本项目组成员个人素质的共同发展，同时也为了能够按时按质完成本项目，特制定本守则。

1．文档化

强调把一切可以用文字表达的内容文档化。

无论是进行软件分析，还是用户调查，还是在软件开发的其他阶段都必须注意写文档，并且以文档为最后输出。因此在软件开发的各个阶段，要求项目组成员把对各个软件的分析心得、用户的一些需求等及时地写在文档中。这些也要作为原始资料保存和项目组的共同财富保存。

2．共同学习

- 代码共享

建立公用的类库。建议每人每半天提交一个公共类库，放在指定机器上的共享目录中。内容包括*.aspx、*.cs 和 readme 以及该类的使用例子。

- 文档共享

要求大家把对系统代码的分析成果，以及在平时学习中得到的各种心得写成文档，贴在 notes 上。

3．学习交流

在学习和工作过程中，经常性的互相交流对于项目组成员的共同发展有不可忽视的作用，因此强调大家在平时的工作、学习中把一些工作的心得、体会互相交流；遇到困难，及时求助别人。

- 值日活动

由项目组成员共同商定：定期举行一次活动（暂定每天一次），由本项目组成员轮流负责。

常规顺序为：蒋 xx、刘 x、张 x、王 xx。

有关事项如下：

- ➢ 时间在一天内由负责人自由选择。初步计划在中午吃饭时间。
- ➢ 内容围绕界面系统以及编程知识、实用工具、其他有关知识等。
- ➢ 值日人负责提前通知大家时间和会议内容，并召集主持会议。
- ➢ 形式不限，可以自己讲，也可以让大家讨论（但自己关于此问题必须要有比较充分的准备）等。
- ➢ 会议结束后，负责把相关资料整理。

- 日常交流

在平时的工作、学习的过程中，应努力创造一种良好的气氛，互相交流、共同进步。

4．编程规范

参见“项目组编程规范”、“项目组界面设计规范”、“项目组 SourceSafe 使用规范”。

5．软件质量管理

基本依据是 ISO9001 质量体系。

要求在软件开发的各个阶段都参照 9001 质量体系完成，形成必要的文档，并把文档保存在指定地方。

具体的质量文档工作主要由支持经理完成。

6．项目管理

本项目组实行按“日”分派任务的原则，具体为：

每日中午开项目组例会，例会内容主要有：

（1）项目组各个成员介绍自己前一天任务的完成情况。

（2）由开发经理安排本日项目组各个成员的任务。

每天，项目组每个成员每天任务的完成情况（百分比）的平均值作为该项目考评的基数。例如，一个成员在该项目每天完成任务的情况为（90%、100%，110%，140%），则该项目其完成任务情况为其平均值110%。（超过100%表示他不仅完成了分配的任务，而且完成了项目组其他有关的任务）

对于能够按时按质完成任务，以及能够承担困难任务等表现突出的成员，报请上级予以升职加薪。

7．计划总结

- 项目组日工作计划、总结。

开发经理每日提交当天工作总结和第二天工作计划。工作总结格式采用原模板。

- 个人计划、总结

提倡项目组每个成员每日作计划总结。每天早上上班，用5~10分钟时间简要安排一下自己当日的工作；在下班前用5~10分钟总结一下自己当日任务的完成情况。

- 阶段性工作计划总结。

主要是在软件开发的各个阶段，开发经理要制定一个比较合理的进度计划，并和大家共同修改、商定。

同时在项目终、项目初等时期也要进行阶段性总结。

8．个人学习

鼓励大家努力学习各种专业知识，不断提高自己的专业水平。

在工作时间，项目组成员的学习都必须围绕整个项目进行，学习和本项目有关的知识。对于和本项目没有关系的学习内容，建议项目组成员在业余时间学习。

9．日常行为规范

基本条例参见“员工行为规范”，同时有适当的补充。

- 禁止在上班期间通过任何方式任何时间聊天、玩游戏。包括用电话、OICQ 等方式聊天。
- 禁止先打上班卡，后长时间离开的行为；更禁止代打卡的行为。
- 除中午时间（12:00—1:30）外，原则上打上班卡后不能上网浏览和工作无关的内容。
- 在任何情况下，都不能因为自己上网、玩游戏而影响其他人的工作、学习。

一旦发生以上情况，除影响当天考评外，还将报请上级处理。后果严重者，撤销在本项目组的一切职务，并且可以取消年假等待遇。

10．体育锻炼

由于软件开发的工作比较累，加上工作性质缺少运动。希望大家都能在工作之余积极锻炼身体，能有一个强壮的体魄。

由项目组支持经理负责安排每天活动一次，形式不限，来督促大家锻炼（初步计划大家每天上下班都步行！）。

11．其他活动

为了活跃大家的气氛、丰富大家的生活，项目组不定期举行一些其他活动。项目结束后，举行庆功宴，大家一起尽情！具体内容由成员申请，大家商定。

本守则在 XXX 项目组内执行。

23.2 C#编码规范

目标：

- 安全：代码完成所需的功能之余，不要产生负作用，即要稳定可靠。
- 易读：类、实例、成员变量、成员函数的命名一目了然。
- 美观：尽量统一项目组内人员的编程风格。

23.2.1 命名

1. 命名原则

（1）所有的函数（变量/类/文件名）应该代表其实际的作用，应该使用有意义的单词或多个词组合，但不要使用人名、项目组名。

（2）所有的函数（变量/类名）一律使用英文。

（3）使用多个单词时不需要使用连线（如下画线），但对于全部大写的宏需要使用连线。

（4）多个词组合较长时，可以使用单词的缩写。

（5）不得使用非常相近的名字类表示几个不同含义的函数（变量/类）。

（6）命名时请考虑名字的唯一性和含义的准确性。

（7）使用项目组专用词汇来表达特定的含义（概念），不得把专用词汇挪作他用。

2. 变量的命名

原则：使用匈牙利命名法命名变量

（1）变量名一般由“类型修饰＋代表变量含意的英文单词或单词缩写”等部分组成。

类型修饰（小写字母）：

n: int，l:　LONG/long， s: short，u: UINT，f:　float

b: bool，by: BYTE，ch: char，　sz: char[],str: string

（2）针对异常捕获过程中的 Exception 变量命名，在没有冲突的情况下，统一命名为 e；如果有冲突的情况下，可以重复 e，比如：ee。

3. 函数的命名

（1）使用动宾词组表达函数实际所做的事。

（2）同名的函数（重载函数）在功能上应该完全相同，在参数上的差别也应一目了然。

（3）不得出现名字非常相近但功能不同的函数。如 CreatePage1()，CreatePage2()等。

4. 类命名

（1）名字应该能够标识事物的特性。

（2）名字尽量不使用缩写，除非它是众所周知的。

（3）名字可以有两个或三个单词组成，但通常不应多于三个。

（4）在名字中，所有单词第一个字母大写，缩写都要大写。

（5）不要使用下画线字符（_）。

5．控件命名规则

控件命名=Web 控件缩写前缀+“_”+变量名

控件	Label	TextBox	Button	ListBox	DropDownList	等等
缩写	lb_XXX	tb_XXX	Btn_XXX	Lb_XXX	Drd_XXX	XXXXX

6．文件命名

（1）文件起名要有实际意义。

（2）源文件应尽量使用 8.3 格式，文件名只能包含字母、数字和下画线，不得使用其他的字母。超长的文件名应使用缩写方式减少文件名的长度。建议使用如下的缩写的规则（部分情况可以有例外，视具体情况而定）：

- 缩写一般可以去掉元音字母以及不发音字母。
- 单词的首字母一般应该保留。
- 当一个单词必须缩成一个字母时，应该选用最有代表性的字母，或首字母。
- 多音节的单词可以去掉后面的音节而只保留前面的第一、二音节。较短的二音节单词一般不缩写，如果必须缩写，可以只保留一个字母。
- 多个单词组成文件名时，应该保留较重要的有意义的单词（或多留几个字母），其他单词可以使用简写或只用首字母，去掉不必要的无意义的单词。
- 可以使用一些谐音表示一个单词，如 2 表示 to，4 表示 for 等，当单词数量少，字母少的情况下，不需要缩写。
- 较短的单词一般不缩写，或缩写为一到两个字母。

23.2.2　代码格式书写规范

1．基本格式

（1）所有的缩进 Tab 键为 4 个空格，每个单词的首字符大写，其余小写。

（2）在代码中垂直对齐左括号和右括号。例：

```
if(x==0)
{
    Response.Write("用户编号必须输入！");
}
```

不允许以下情况：

```
 if(x==0) {
Response.Write("用户编号必须输入！");}
```

或者：

```
 if(x==0){Response.Write("用户编号必须输入！");}
```

（3）在大多数运算符之前和之后使用空格，这样做时不会改变代码的意图却可以使代码容易阅读。例：int j = i + k; 而不应写为：int j=i+k;

（4）编写 SQL 语句时，对于关键字使用全部大写，对于数据库元素（如表、列和视图）使用大小写混合。

（5）将每个主要的 SQL 子句放在不同的行上，这样更容易阅读和编辑语句。

2．注释的写法

（1）在你劳神的地方请加上详细的注释说明。除了最简单的存取成员变量的 Set_/Get_成员函数之外，其余大部分的函数写上注释是良好的习惯。尽量使你的程序让别人很容易看懂。

（2）太多的注释会使程序很难看，但一些复杂的算法和数据结构处还是要加上注释的，这样别人就容易看懂。否则时间长了，你自己都未必看明白了。

（3）如果是对某一段程序（算法/结构）的注释，在程序头直接用//再空一格来进行说明，一行不要超过 80 字符。

（4）为了防止在阅读代码时不得不左右滚动源代码编辑器，每行代码或注释不得超过一显示屏。

（5）使用连续的多个//表示注释行（不要超过 80 字符）。

（6）文件头部应有注释，简单描述文件的内容。

（7）对于程序中的比较关键的算法和函数，必须加注释。

3．cs 文件的书写

（1）各个部分应使用注释行和空行分割，并在必要的地方写上注释。

（2）函数之间用注释行和空行分割。

（3）重要的函数在函数头部加上注释。

（4）{ 和 }分别单独占用一行，且上下对齐，中间的内容缩进一个 Tab。

23.2.3　其他

1．变量

（1）float 和 bool 禁止用"=="判断。bool 应该用逻辑运算关系符，而 float 应该用差值区间来判断“相等”。

（2）类型转换一律用显式类型转换。

（3）类型的长度一律用 sizeof()获得。

（4）当声明一个变量时，务必要自己初始化一下变量。

2．函数

（1）功能要单一，函数名要名副其实。

（2）要易懂，实现时不要过分追求技巧，优化放到后面去做。

（3）长度一般禁止超过 200 行。

（4）要检查输入值是否合法。实现（成员）函数时务必要求输入参数是在要求范围之内，尤其你定义的（成员）函数给别人调用时，要判断其合法性。

（5）调用函数时要严格按照接口规范调用，调用后要判断执行情况，并做适当的错误处理（稍后会给出错误和异常处理规范）。

（6）尽量避免整块复制代码段，如果出现这样的情况要分析原因，如果这段代码完成独立的功能，应考虑使用函数，否则，应考虑使用宏定义。否则因为修改引起的不一致往往是错误的根源。

（7）除极其简单的函数外，其他的函数在入口处必须加上 FMAT_TRACE()，参见错误和异常处理规范。

（8）函数的出口尽量唯一，最好在出口处加上 FMAT_TRACE()。

（9）写代码时，尽量减少堆的分配次数，能使用 Stack 的尽量使用 Stack。

（10）函数编写必须精练，消除冗余的代码，删除不用的变量。

（11）if/while 等语句中的条件表达式的运算结果必须为显示的 BOOL 量。

（12）禁止用 goto 语句。

23.3 SourceSafe 使用规范

版本控制是工作组软件开发中的重要方面，它能防止意外的文件丢失、允许反追踪到早期版本、并能对版本进行分支、合并和管理。为了对源代码进行更好地管理，要求项目组内的人员掌握 SourceSafe 的使用。为了能充分利用 SourceSafe 的功能，实现源代码的有效管理，减少冗余、提高编码效率，特指定项目组的 SourceSafe 使用规范。

1．使用规定

（1）只加入（Check in）建造（Build）你的工程所必需的文件和相关设计文档，不要将中间文件加入到 SourceSafe 中。（只加入 cs，Global.asax，Web.config，csproj，sln，aspx 等，不加入 dll，pdb，等文件）。

（2）除了资源使用的 bmp，ico，gif，jpg 等文件，或者自己加入的 lib 库外，不允许把二进制文件或者很大的文件放入 SourceSafe，以提高整个效率。由于图片文件的修改不可以保存在 SourceSafe 所以一般情况下尽量不要将 bmp，ico，gif，jpg 等文件 Check in。

（3）cs 文件不要多人同时 Check out，在 Checkout.cs 文件前，一定先 Get Latest Version。在打开 cs 之前，一定先 Check out。

（4）为了避免多人同时对一个程序进行修改，造成多次覆盖，所以在对资源进行编辑前应该首先 Check out。

（5）为了避免影响他人使用，保持 Check out 的时间不要超过当天。尽量编辑完成调试正常后立刻 Check in。较重要的公用文件保持 Check out 的时间应尽量短。

（6）自己应该保证自己代码的正确性，只能把编译通过的代码 check in 进数据库；应该信任别人的工作，对别人的代码的修改必须争得别人的同意。不允许 Check in 编译未通过的代码，或有严重错误无法运行，影响他人工作的代码。

（7）Check in 后，检查一下服务器上的代码是否正确。（比较一下本地的代码与服务器上代码的差别，比较一下最近两个版本之间的差别。）

（8）组员尽量在当天下午将当天调试通过的文件进行 Check in。

（9）提倡使用 SourceSafe 的“自动合并”功能。

（10）良好的习惯是在修改之前首先 Check out，然后再 Get Latest Version（取出最新版本），在 Check in 之前，保证自己的修改正确。

（11）一般使用.NET 和 SourceSafe 结合的方式。

2．使用注意事项

（1）SourceSafe 中的源程序应保证完整性，即工程所需的文件应该全部放在 SourceSafe 中。在任何的机器上通过 SourceSafe 可以很快重建工程。

（2）注意 csproj 文件的正确性：csproj 文件中包含的所有的文件都应该在 SourceSafe 中，且 csproj 关联的源文件是完整的。通过 Get SourceSafe 中的 csproj 文件后，在 Get csproj 下面的所有关联文件，可以完整 Build，不需要单独 Get 其他的文件。

（3）注意 csproj 文件的紧凑性，csproj 中不应保留未使用的源文件。

（4）在 SourceSafe 中，一个工程只需要设置一个 Working Directory，而其所包含的子目录和子工程不需要单独设置 Working directory，保证个人本地的目录结构都是一样的。

3．完整性的检查方法

（1）关闭.NET，修改本地的工作目录名。

（2）启动.NET，在 SourceSafe 下重建工程（打开 SourceSafe 下的 sln 或 csproj 文件）。

（3）在.NET 中，Get 所有文件。

（4）检查编译连接是否能通过。

4．工程的 Build 和 Label

一般我们会定期进行 Build，提供测试和最终发布所要的二进制代码。为了保证可追踪性，一般在 Build 前 Get 所有的源文件，Check out 资源文件，修改其中 Version 资源中的 Build 号后 Check in。开始编译。然后在 SourceSafe 中使用 Build 号和编译日期作一个 Label，以便以后恢复和再现错误使用。如果编译出现问题，进行修改后，必须重新作 Label。

主要是要保证编译所用的源文件必须与编译出来的二进制文件完全对应（因此 Get 后必须马上作 Label）。如果保证了这一点，在以后的工作中会非常有好处。比如，你想知道从某一个版本（L5.0.0.30）以后的所有改动，或者你想还原 L5.0.0.30 对应的源文

件，以便重新编译和再现用户的问题等。

23.4 数据库命名规范

1. 数据库文件命名

数据文件命名采用系统名+_+文件类型，比如系统名为 ComSys，则数据库文件命名为 ComSys_database.mdf，有的数据库文件有多个，比如 SQL Server 就有 2 个，一个是数据库文件，另一个是日志文件，那么他们的文件命名分别为 ComSys_database.mdf，ComSys_log.log。

2. 数据库对象命名规范

数据库对象包括表、视图（查询）、存储过程（参数查询）、函数、约束。对象名字由前缀和实际名字组成，长度不超过 30。

（1）前缀

使用小写字母。如：

```
表          tb
视图        vi
存储过程    sp
函数        fn
```

（2）实际名字

实际名字尽量描述实体的内容，由单词或单词组合，每个单词的首字母大写，其他字母小写，不以数字和_开头。如：

```
表          User_Info
视图        User_List
存储过程    User_Delete
```

（3）示例

合法的对象名字类似如下：

```
表          tb_User_Info    tb_Message_Detail
视图        vi_Message_List
存储过程    sp_Message_Add
```

3. 表的命名

（1）表名由前缀和实际名字组成。前缀使用小写字母 tb，表示表。

对工作用表来说，表名可以加上前缀WORK_后面附上采用该表的应用程序的名字。在命名过程当中，根据语义拼凑缩写即可。注意，由于 ORCLE 会将字段名称统一成大写或者小写中的一种，所以要求加上下画线。

举例：

定义的缩写 Sales: Sal 销售；

Order:Ord 订单；

Detail:Dtl 明细；

则销售订单明细表命名为：tb_Sal_Ord_Dtl;

（2）如果表或者是字段的名称仅有一个单词，那么建议不使用缩写，而是用完整的单词。

举例：

定义的缩写 Material Ma 物品；

物品表名为：tb_Material，而不是 tb_Ma。

但是字段物品编码则是：Ma_ID；而不是 Material_ID

（3）所有的存储值列表的表前面加上前缀 Z

目的是将这些值列表类排序在数据库最后。

（4）所有的冗余类的命名（主要是累计表）前面加上前缀 X

冗余类是为了提高数据库效率，非规范化数据库的时候加入的字段或者表。

（5）关联类通过用下画线连接两个基本类之后，再加前缀 R 的方式命名，后面按照字母顺序罗列两个表名或者表名的缩写。

关联表用于保存多对多关系。

如果被关联的表名大于 10 个字母，必须将原来的表名的进行缩写。如果没有其他原因，建议都使用缩写。

举例：表 Object 与自身存在多对多的关系，则保存多对多关系的表命名为：tb_R_Object；

表 Depart 和 Employee 存在多对多的关系；则关联表命名为 tb_R_Dept_Empl。

4．字段命名

（1）采用有意义的列名，表内的列要针对键采用一整套设计规则。每一个表都将有一个自动 ID 作为主健，逻辑上的主健作为第一组候选主健来定义，如果是数据库自动生成的编码，统一命名为：ID；如果是自定义的逻辑上的编码则用缩写加“ID”的方法命名。

举例：销售订单的编号字段命名：Sal_Ord_ID；如果还存在一个数据库生成的自动编号，则命名为：ID。

（2）数据库字段名全部采用小写英文单词，单词之间用“_”隔开，命名规则是表别名+单词，如：user_name，user_pwd（表别名规则，如果表名是一个单词，别名就取单词的前 4 个字母；如果表名是两个单词，就各取两个单词的前两个字母组成 4 个字母长的别名；如果表的名字由 3 个单词组成，则从头两个单词中各取一个然后从最后一个单词中再取出两个字母，结果还是组成 4 字母长的别名）。

5．视图的命名

（1）视图名由前缀和实际名字组成，中间用下画线连接。前缀使用小写字母 vi，表示视图。

合法的视图名类似如下：

```
vi_User
vi_UserInfo
```

（2）命名应尽量体现各视图的功能。

6．触发器的命名

触发器以 tr 作为前缀，触发器名为相应的表名加上后缀，Insert 触发器加“'_I”'，Delete 触发器加“'_D”'，Update 触发器加“'_U”'，如：tr_Customer_I，tr_Customer_D，tr_Customer_U。

7．索引命名

索引名= idx +“_”+ 表缩写+相关字段/索引含义。

8．存储过程名

存储过程名由前缀和实际名字加操作名字组成，中间用下画线连接。

前缀：使用小写字母 sp，表示存储过程。

操作名字：Insert|Delelte|Update|Caculate|Confirm

合法的存储过程名类似如下：

```
sp_User_Insert
```

9．变量名

变量名采用小写，若属于词组形式，用下画线分隔每个单词，如@sys_err_no。

10．命名中其他注意事项

（1）以上命名都不得超过 30 个字符的系统限制。变量名的长度限制为 29（不包括标识字符@）。

（2）数据对象、变量的命名都采用英文字符，禁止使用中文命名。绝对不要在对象名的字符之间留空格。

（3）小心保留词，要保证你的字段名没有和保留词、数据库系统或者常用访问方法冲突。

（4）保持字段名和类型的一致性，在命名字段并为其指定数据类型的时候一定要保证一致性。假如数据类型在一个表里是整数，那在另一个表里可就别变成字符型了。

23.5 用户界面设计规范

1．菜单

（1）所有的菜单项后面都应有快捷键。快捷键用“()”表示而不用“[]”。

- 例如：“文件”后的“(F)”。
- 定义快捷键的原则是不冲突时使用英文缩写的第一个字母，有冲突时依次使用第二、第三个字母。

（2）选择后要弹出对话框的菜单项要在菜单项文字后面加“...”标明。

（3）菜单命令用语原则上使用动宾词组。在确保语意明确的基础上尽量简短。

（4）各个子模块的菜单项的布局和风格应保持一致。

- 各个子模块各自的数据处理命令集中在“编辑（E）”菜单下。
- 各个子模块各自的数据打开、存储等命令集中在“文件（F）”菜单下。
- “显示（V）”菜单下的菜单项：各个子模块的显示设置命令。各个子模块的工具

条显示命令称为“工具条（L）”，而不再称为“描上工具条”等。

- 有工具箱的子模块，统一在“编辑（E）”菜单下设置一个“工具（T）”二级菜单，对应工具箱的命令菜单项集中在该二级菜单下。
- 工具需要设置选项的子模块，统一在“编辑（E）”菜单下设置一个“设置工具参数（O）”二级菜单，对应工具选项的命令菜单项集中在该二级菜单下。
- “显示（V）”菜单下有显示比例设置时，使用“缩放（F）”二级菜单。显示比例集中在二级菜单下，放大比例在上，缩小比例在下。显示比例统一用百分比表示。
- “编辑（E）”菜单下的“撤销（U）”命令应在撤销具体操作时标明具体撤销内容。例如：“撤销（U）编辑”。

（5）弹出菜单。辅助菜单必须在可视化界面上拥有对应的按钮或者菜单选项。由于辅助菜单由用户单击鼠标左右键或者别的动作才能调出来显示给用户。无法清晰的显示给用户，所以对应选项应该可以通过别的途径得到，例如界面上有相应控件或弹出右键菜单的按钮等。

2. 对话框

（1）对话框上的控制的 Tab 顺序应一致。一般为从左到右，从上到下。

（2）除非特殊情况，“按钮”和“Edit”控制使用系统的默认大小。如果要改变，一般只改变宽度不改变高度。

（3）通用按钮的汉化命名应统一。

- “OK”和“CANCEL”按钮统一称为“确定”和“取消”。
- “Browse”（选择路径名）按钮统称为“浏览（B）”。

（4）除了“确定”和“取消”按钮外，所有的按钮文字上都应有快捷键。

（5）按钮在对话框中的位置一般为右边，下边或右下方。

（6）“Edit Box”和“ComboBox”控制前的前导性文字后面要加“：”。

（7）需要用户输入的数值带单位时，如果使用“Edit Box”就在其后加表示两词的“Static”文字或可以选择的单位的“ComboBox”，用户输入的数值不带单位；如果使用“ComboBox”，就把单位和数值一起放在 ComboBox 内。

（8）一般情况下使用“列表框”代替“Ratio Button”控制。

（9）会改变但不需用户输入的文字一般用加边框的“Static”控制或 Read only 的“Edit”控制。

（10）只允许输入整数的“Edit”控制在编辑资源时加上“Number”选项。

（11）只允许输入整数而且数值有范围的“Edit”控制应使用主框架的 EasyNumEdit 控制。

（12）使用“Spin”控制辅助用户输入整数时，“Spin”控制应放置在“Edit Box”内。

（13）对话框的 Title 应使用简明的动宾词组以表示其用途。

（14）所有对话框都应有“帮助（H）”按钮。

（15）所有对话框都应保留 Title 上的关闭按钮，设计资源时保留“system Menu”选项。

（16）各个子模块有长时间操作时，统一使用进度对话框方式。

- 不允许用户终止时，使用无 Title 的对话框，进度条上面设置解释性文字。
- 允许用户终止时，使用有 Title 的对话框，Title 里设置解释性文字。进度条右面设置百分比数值表示进度，进度条下面设置一个“取消”按钮，按钮位置居中。
- 解释性文字的格式是：“当前正在××××，请稍侯...”，左对齐放置。
- 进度条的高度统一定为 12（Resource Editer 单位）。

（17）公用对话框的风格尽量一致。

3．工具条和工具箱

（1）一般使用彩色按钮。

（2）工具条或工具箱上有的按钮，菜单上都应有对应菜单项。

（3）各个模块的按钮图案风格一致。

（4）工具条必须 Docking，工具箱必须 Floating。不允许两种状态之间的转换。

（5）各个子模块的工具参数设置范围应统一。

- 铅笔直径的范围为 1~64。
- 允许用户选择是否自动补点时，同时应允许用户选择补点间距。

4．快捷键

（1）在菜单及按钮中使用快捷键可以让喜欢使用键盘的用户操作得更快一些。

（2）快捷键应该遵循用户的常用习惯，并和大多数软件（如 Office）保持一致。如“Ctrl+D”为删除、“Ctrl+F”为查找、“Ctrl+H”为替换、“Ctrl+I”为插入、“Ctrl+N”为新建、“Ctrl+S”为保存、“Ctrl+O”为打开。

5．状态条（StatusBar）

（1）各个子模块在保留主框架状态条的项目的基础上可以添加各自的项目。

（2）主框架在状态条上设置一个时钟（Timer）。

6．资源

一个多姿多彩的人机交互界面，少不了精美的鼠标光标、图标以及指示图片、底图等。这些资源的使用遵守以下规则。

（1）有标准的图标风格设计，有统一的构图布局，有统一的色调、对比度、色阶，以及图片风格。

（2）底图应该融于底图，使用浅色，低对比，尽量少的使用颜色。

（3）图标、图像应该很清晰的表达出意思，遵循常用标准，或者用户机器容易联想到的物件，绝对不允许画出莫名奇妙的图案。

（4）鼠标光标样式统一，尽量使用系统标准，杜绝出现重复的情况。

7．字体

（1）使用统一字体，字体标准的选择依据操作系统类型决定。字体大小根据系统标准字体来，例如 MSS 字体 8 磅，宋体的小五号字（9 磅）五号字（10.5 磅）。

（2）需要使用自定义字体时，统一使用“Arial”字体（输出汉字时自动使用系统默认的汉字字体）。

（3）各个模块的自定义字体的大小按用途统一，标尺统一、ComboBox 统一等。统一使用主框架定义的常量。

（4）所有控件尽量使用大小统一的字体属性，除了特殊提示信息、加强显示等例外情况。

8．出错信息和警告

出错信息和警告是指出现问题时系统给出的坏消息，对于出错信息和警告应该遵循

以下原则：

（1）信息以用户可以理解的术语描述；

（2）信息应提供如何从错误中恢复的建设性意见；

（3）信息应指出错误可能导致哪些不良后果，以便用户检查是否出现了这些情况或帮助用户进行改正；

（4）信息应伴随着视觉上的提示，如特殊的图像、颜色或信息闪烁；

（5）信息不能带有判断色彩，即任何情况下不能指责用户。

23.6 安装盘制作规范

（1）发行的产品光盘应该具有 Autorun 的功能。在 Autorun 画面上，要求出现本企业的注册商标（英文版采用英文商标），如果属于电子出版系统类的软件则要求采用驰名商标；Autorun 画面中要求出现下列通用选项："浏览此光盘"、"安装说明"、"退出"，其余选项由项目组自行决定；光盘 Autorun 功能也可跳过 Autorun 画面，直接启动标准安装程序。

（2）光盘要求有光盘图标（即.ico 文件），项目组可以自行采用与产品相关的.ico 图案，也可以采用公司提供的通用.ico 图案。

（3）光盘卷标（disk volume）命名规定为：公司缩写_软件名称。（这里软件名称一般指软件英文名称，如没有英文名称则采用汉语拼音，如多个软件合刻在一张盘，建议卷标名直接为公司名称缩写）。

（4）光盘要求包含 license.txt 文档，该文档为整个公司产品统一文档。

后　　记

程序员速成的五条“真经”

1．掌握核心——以“不变”应“万变”

软件技术的发展可谓日新月异，曾几何时笔者还在拿着 Turbo C 闯荡天下，但眨眼间已经是 Java 和.NET 的天下了；2002 年微软首次推出.NET 架构、VisualStudio.NET2009 的那一刻，仿佛就在昨日，而如今程序员都在使用 VisualStudio. NET2008 做开发了……

曾经有人说，程序员是淘汰最快的职业，很多人对此不置可否。的确，作为一个程序员，要一直站在技术的前沿，那需要很大的毅力和挑战。我们该如何应对这种挑战呢？

其实，细心的程序员分析一下，可以看到：**在如此快速的技术变化后面，有一些东西变革缓慢或者基本保持不变——这就是软件开发的流程和项目相关技术！**原来困扰我们的只是一个开发工具、一门编程语言而已。真正的武学大师，何必在意手中的武器到底是刀还是剑呢？

所以，笔者认为，了解软件开发的过程，把握各个阶段需要掌握的基本知识，熟悉各个阶段所涉及的工具，这才是“以不变应万变之道”！这些才是真正的“九阳真经”！

掌握了这些，所有的编程语言和开发工具，任君选择。是用 Java 这个“倚天剑”，还是用 C#这把“屠龙刀”，就由我们的个人爱好和开发需要来定了！

变的是版本，不变的是流程。只有以“不变”应对“万变”，我们才能在程序员这条路上走得更远。

因此，在《软件开发这点事儿》这本书中，笔者特别讲解了软件开发过程中所涉及

的相关技术、工具，希望大家能够快速掌握这些知识，真正踏入软件开发的殿堂！

2．软件开发中的二八原理

在软件编程中，同样有二八原理的存在。

首先，**在编程环境提供的所有类库中，常用的类只占 20%**。

无论是 VC++，还是 Java、.NET，都提供了大量的类库，这是现代编程技术发展的趋势。通过这些类的调用，可以使得程序员的编程效率大幅提高，同时能够编出功能更加强大的程序。这些类库的重要性是毋庸置疑的。不信，你用 Turbo C 编写一个邮件发送程序试试？呵呵，很难，几乎不可能；而用 Java 或.NET 类库呢，几行代码而已。

有的学员，曾经来信问到，面对类库中动辄成百上千个类，我们该如何学习呢？难道要去死记硬背这些类的每个属性和方法吗？

对于此问题，笔者的回答是：熟记、掌握最常用的类就可以了，而对于其他类，我们要做到的是：遇到相关问题时，能记起类库中有相应的类，然后去查询帮助使用即可。

然而，到底哪些是常用的类呢？大家不妨比较一下 MFC 类库、JDK 类库、.NET 类库，可以发现它们的类库分类有很多相似之处，也有很多功能类似的类。笔者认为，不论哪种语言，以下一些类都是需要掌握的，比如界面基础控件类、文件 IO 类、图像处理类、多线程类、数据库类等。

其次，**每个开发工具如 Visual Studio.NET 或 Eclipse，常用的功能只占 20%**。

当今社会，各种编程工具如同照相机等数码产品一样，功能变得越来越强大，同时操作上越来越傻瓜化。在开发工具繁多的功能中，实际上常用的功能不足 20%。因此，对于每种开发工具，我们仅仅重点掌握这些常用功能就足够我们进行日常开发了。

典型的常用功能有项目文件的操作功能、与编程相关的各种文件、数据库的添加功能、调试功能、项目属性设置功能、开发环境个性化设置功能等。

3．基于问题的学习方式

在笔者近十年的教学生涯中，最喜欢的是 PBL（problem based learning）学习方法。该方法强调以问题作为学习的起点，所有的问题来自于实际的应用。实际上，软件本质上也是用来解决用户问题的一种工具和手段，每种知识点也是也源于实际的应用，没有实际应用的知识在编程中是毫无用处的。

因此，强烈大家带着问题去学习。

同时，要培养自己的解决问题能力。正如在前文所述，什么是编程高手？是熟悉所有.NET 类库或 JDK 类库？显然不是。现实中可能有这样的“高手”，但寥寥无几。**真正的高手是在遇到各种问题后能很快地找到解决问题的方法，即解决问题的高手**。

也许有的读者要问，我什么都不了解呢，问题从哪里来？当然是从实践来，这也是我们接下来要讨论的话题。

4．实践是最好的老师

经常有学员来信问，“我是一个.NET 初学者，请问我该如何学习.NET 呢？”。这个问题涉及了编程语言的学习方法。

笔者认为有两种学习路线：

其一：理论学习路径法。这种方法是大多数院校采用的教学方式，把编程语言当做一门学科来学习，学习的目的更多地是为了考试。笔者曾经在招聘市场上遇到很多刚毕业的计算机专业的大学生，问“学过.NET 吗”，答“学过，不仅学过.NET，还学过 Java 呢！”，又问“用.NET 做过实际的项目吗？”，答案往往是“没有”。这也算是一种悲哀吧。怪不得所有招聘人员都希望要有工作经验的人呢。但，请注意，他们要的实际不是“工作经验”，而是“项目经验”，白话讲是“干过活没有”。如果大家能够在大学期间就抽空积累一些项目经验，何愁找不到好的工作呢？

其二：实用导向路径法。即面向实用。就如同一个痴心学武的人，学再多的招式，不去实际找人演练，永远成不了武林高手一样；要想真正地掌握一门编程语言，必须和实际应用结合。**在应用中，产生问题，带着问题去学习，学习后进行应用，这是一个不断进步的循环过程。**

显然第二种方法更适合软件编程的学习。1997 年笔者开始接触 VC++的学习，看了很多的书、学了很多的理论知识，但一直觉得编程水平没有进展。后来有机会接手一个学籍管理系统的项目，直到系统开发完成后，才觉得自己真正迈入了 VC++的门槛。

5．经验的积累和技巧的总结

真正的程序员是善于总结的。随着项目经验、编程经验的增加，每个程序员都会积累很多的编程技巧和心得。记得一定要把这些技巧和心得用文件的形式保存下来，这是你个人的财富！

6．关于《.NET 编程这点事儿》

基于以上的一些感悟，和自身近 15 年来学习编程的经验和苦处，笔者计划近期推出《.NET 编程这点事儿——.NET 编程工具手册》一书。这本书是笔者上述思想的具体

体现，它具有以下一些特点：

（1）问题集。这本书可以说是一本.NET 学习的问题集，收录了读者在.NET 学习中常遇到的大多问题，能帮助读者快速上手。

（2）实用导向。该书以实际项目应用为主线，以面向应用、提高动手能力为导向，以项目实践、融会贯通工作经验为重点，以培养实际专业能力为目标。

（3）经验技巧集。本书侧重的是.NET 编程具体知识在实际项目中的应用技巧，重点在于技能的归纳、总结和经验分享，并提供大量实用技巧代码和模板。

本书将在 2009 年年中和大家见面，由电子工业出版社出版，详细信息欢迎大家随时关注 www.broadview.com.cn 上的新书预告。希望此书的出版能够有助于广大.NET 编程爱好者的学习，帮助大家顺利走入编程之路！

邵志东

精品图书免费阅读！

倾情奉献数载开发经验，丝丝入扣讲述 ASP.NET 控件开发底层原理...

《庖丁解牛——纵向切入 ASP.NET 3.5 控件和组件开发技术》

电子工业出版社 【书号】9787121079795 2009 年 3 月 出版 含光盘 2 张

网络购买地址：http://www.china-pub.com/192102

内容介绍：本书主要介绍 ASP.NET 的控件开发，书中通过 70 多个例子讲解了 ASP.NET 控件开发技术的各个方面，而且剖析了很多控件中系统基类源代码，读者从这些系统源代码可以体会设计模式思想。如果扎实地掌握了 ASP.NET 控件的运行机制，开发一个页面级的 ASP.NET 应用程序会变得非常简单。本书宗旨就是让开发人员真正理解 ASP.NET 技术，帮助开发人员提高 ASP.NET 开发的技术水平。学完本书后您不仅能够掌握控件开发各个方面的技术，而且深晓 ASP.NET 的工作原理。对大部分使用 ASP.NET 技术开发两年左右的开发人员来说，本书 75%以上的内容可能没有接触过，本书将通过笔者精心准备的一些生动的例子来阐述这些内容。

本书适合从事 ASP.NET 技术多年想往深度理解 ASP.NET 工作机制的开发人员阅读，对于刚刚入门的初级开发人员，本书的前面几章也介绍了几个入门性的应用例子，本书还可以作为在校学生的就业预备教材。针对读者安装的开发环境不同，笔者分别为 VS 2005 和 VS 2008 建立了解决方案，即无论您是使用 VS 2005 还是 VS 2008 都可以成功打开本书随书光盘的代码，只是使用 VS 2005 打开的解决方案不包含.NET 3.5 对应章节内容的示例代码。

《庖丁解牛——纵向切入 ASP.NET 3.5 控件和组件开发技术》精彩片断试读

客户端回传事件接口 IPostBackEventHandler

要使控件捕获回发事件，控件必须实现 System.Web.UI.IPostBackEventHandler 接口。此接口约定允许控件在服务器上引发事件来响应来自客户端的回发。IPostBackEventHandler 接口包含一个方法。

```
/// <summary>
/// 获得本书更多内容,请看:
/// http://blog.csdn.net/ChengKing/archive/2008/08/18/2792440.aspx
/// </summary>
public interface IPostBackEventHandler
{
    void RaisePostBackEvent(string eventArgument);
}
```

参数 eventArgument 表示要传递到事件处理程序的可选事件参数，一般通过此参数可以确定不同的引发事件源，进而作不同的逻辑处理。在本章最后有个例子说明 eventArgument 参数用法。回发后，页框架就会搜索发送的内容，并确定发送的名称是否与实现 IPostBackEventHandler 的服务器控件的 UniqueID 对应。如果对应，页框架就会在

该控件上调用 RaisePostBackEvent 方法（在引发更改事件后）。

以下代码片段显示了在服务器上引发 Click 事件的 RaisePostBackEvent 实现：

```
    /// <summary>
    /// 获得本书更多内容,请看:
    /// http://blog.csdn.net/ChengKing/archive/2008/08/18/2792440.aspx
    /// </summary>
    public void RaisePostBackEvent(String eventArgument)
    {
        ButtonEventArgs e = new ButtonEventArgs(eventArgument);
        OnClick(this, e);
    }
```

在该方法中主要完成调用 OnClick(e)事件功能，其中包含两个参数：第一个参数为当前控件本身(即一般事件体中 sender，类型一般为 object)；第二个参数 e 为 ButtonEventArgs 参数类型对象，是继承于 System.EventArgs 类实现的参数类，在该类中可以定义与代码逻辑相关的任意属性，作为事件体的参数。

最后，RaisePostBackEvent 需要被客户端引发才能够执行，下面是一段能够引发服务端事件的代码：

```
    /// <summary>
    /// 获得本书更多内容,请看:
    /// http://blog.csdn.net/ChengKing/archive/2008/08/18/2792440.aspx
    /// </summary>
    protected override void Render(HtmlTextWriter output)
    {
        output.Write("<INPUT TYPE=submit name=" + this.UniqueID +
            " Value='Click Me' />");
    }
```

这段代码输出一个 HTML 的 button 标签，并设置为提交类型。非常重要一点是，不要忘记设置其 name 属性，因为当回发后，页框架就会搜索发送的内容，并确定发送的名称是否与实现 IPostBackEventHandler 的服务器控件的 UniqueID 对应。如果对应，页框架就会在该控件上调用 RaisePostBackEvent 方法；反之，如果没有设置按钮的 name 值为 UniqueID 属性值，当单击按钮时页框架就不会引发该控件的 RaisePostBackEvent 方法，因为只有名称为 UniqueID（服务器控件服务端 ID）的按钮才会被注册为具有 IPostBackEventHandler 接口功能的控件。

下面通过一个简单的完整例子，了解事件回发处理机制，代码如下：

```
    /// <summary>
    /// 获得本书更多内容,请看:
    /// http://blog.csdn.net/ChengKing/archive/2008/08/18/2792440.aspx
    /// </summary>
    [DefaultEvent("Click")]
    [ToolboxData("<{0}:PostBackEventControl
runat=server></{0}:PostBackEventControl>")]
    public class PostBackEventControl : Control, IPostBackEventHandler
    {
        public event EventHandler Click;
        protected virtual void OnClick(EventArgs e)
```

```
        {
            if (Click != null)
            {
                Click(this, e);
            }
        }
        public void RaisePostBackEvent(string eventArgument)
        {
            OnClick(EventArgs.Empty);
        }
        protected override void Render(HtmlTextWriter output)
        {
            output.Write("<INPUT TYPE=submit name="+this.UniqueID+"Value='单击我'
            />");
        }
    }
```

该控件内容比较简单，仅输出一个提交类型的按钮，由于这个按钮类型为 submit，所以当单击按钮时，其本身已经可以提交事件到服务器，但仅仅这样主控件还不能够捕捉到该按钮事件。控件能够捕捉处理该事件需要具备两个条件：第一，主控件继承了 IPostBackEventHandler 接口以及实现了 RaisePostBackEvent 方法；第二，必须有 name 值为 UniqueID 的客户端标签，页框架只认识控件的 name 属性。只有这两个条件同时具备才能够使控件具备捕捉并处理事件的机会。

语句[DefaultEvent("Click")]的功能是定义 Click 事件为默认事件。通常把最常用的一个事件定义为默认事件，如果定义了默认事件，在设计器中双击控件时系统会自动从源代码视图（*.aspx）切换到后面代码（*.cs）页面，并可以自动注册默认事件；否则，仅切换到后台代码，但不注册任何事件。

类代码中最上面的一句：

```
    public event EventHandler Click;
```

这条语句定义了一个委托事件。EventHandler 是一个预定义的委托，专用于表示不生成数据的事件的事件处理程序方法。如果事件生成数据，则必须提供自定义事件数据类型，并且必须要么创建一个委托（其中第二个参数的类型为自定义类型），要么使用泛型 EventHandler 委托类并用自定义类型替代泛型类型参数。

在 5.1.1 节讲过事件的意义及其完成的功能，即在一个类中执行另一个功能类的功能，要想执行另一个类中的某个功能方法，必定要在当前类中保留另一个方法的引用，委托和事件就是为实现此功能而出现的。委托类似于 C++中的函数指针，它一般指向一个方法。这样，当调用本类中的委托就相当于调用了另一个类中的方法。

若要将事件与处理事件的方法关联，请向事件添加委托的实例和委托事件实例，如：

```
    this.Button1.Click +=new EventHandler(Button1_Click);
```

除非移除了该委托，否则每当发生该事件时就调用事件处理程序。

另外，委托的应用比较广泛，还有更加复杂的应用，比如定义一个委托可以同时指向多个事件，即一个委托指定一个事件列表，当调用委托时可以执行一系列的事件。这里限于篇幅就不多讲了，更多关于事件委托概念请查看微软官方文档，那里面已经讲得很详细了。

精品图书免费阅读！

《SilverLight 魔幻银灯》

苏鹏 著　　ISBN 978-7-121-07468-4　　2008 年 11 月出版

定价：49.00 元　　网购地址：www.china-pub.com/129848

内容简介

本书通过翔实的介绍为广大开发者揭开微软 SilverLight 开发的神秘面纱，还配有大量截图，介绍了微软的设计工具 Blend 在设计 SilverLight 方面的使用。本书适合进行网页开发的程序员和页面设计人员，以及对微软 SilverLight 技术有兴趣的 Web 程序设计人员阅读。

本书特色

- 笔者在这本书中尽其所知介绍了 SilverLight 开发的诸多细节，并使用了一些范例让读者能够快速了解 SilverLight 开发的种种方式和技巧。
- 本书通过笔者的收集整理并结合自身的经验，让广大读者在开发中少走弯路，读者通过本书的学习即可快速地了解 SilverLight 开发的全貌。
- 本书尽力介绍了各种开发技术的方式，并介绍了其内在的运转方式，相信聪明的读者可以在阅读中举一反三地创造出自己的应用。
- 本书配有一张光盘，其中包括按章存放的示例程序。

作者介绍

苏鹏，现任中国网通集团北京市分公司软件开发架构设计师，常年从事 ASP、ASP.NET 等 Web 开发工作。于 2006 年荣获微软最有价值专家（MVP）称号，2007 年和 2008 年连任。目前从事的工作中开始关注 Web 表示层设计工作。

《SilverLight 魔幻银灯》精彩片断试读

SilverLight 2.0 中的动画

SilverLight 2.0 中一个非常让人激动的功能就是多媒体的播放和动画效果的展示，这一章我们就来介绍如何实现多媒体和动画效果。

动画效果可以增强应用程序的图形互动性，使用动态的背景颜色或者变化的前台样式可以让用户感觉非常 cool。

动画的基础实现方式说起来比较简单，基本原理就是利用人类的视觉暂留现象，通过快速的切换画面给人以画面运动起来的效果。人眼在某个视像消失后，仍可使该物像在视网膜上滞留 0.1～0.4 秒左右。当我们欣赏电影时，投影机会在屏幕上每秒钟匀速播放 24 格静止的画面。一系列静态画面就会因视觉暂留作用感觉到第一格画面还未消失，第二格画面便出现了，使影像在观看者的视网膜上暂留，造成一种连续的视觉印象，观众便可看到

一组活动的画面，产生逼真的动感，于是屏幕上的影像就好像是连续的。

在实际操作中每秒绘制 24 幅图像其实是相当麻烦的，即使是现在的硬件条件，这种渲染也会耗费大量的 CPU 时间。所以通常情况下，在绘制动画的时候，SilverLight 和 Flash 一样，使用一种叫做定格绘制法的技术来实现。

这种技术起源于电影界，在没有大型计算机渲染技术的时代，绘制一个恐龙大战的场景简直是不可思议的，当然在硬件条件相对落后的时代，不可能使用电脑来实现这个渲染工作。那么这些电影人是如何完成这一艰巨工作的呢？首先，电影人制作一个模型，这个模型是固定的，不会动也不会叫，然后电影人拍一帧照片，把摄影机定格，接着手动去调整一下模型的位置，然后再拍一帧，就这样一帧一帧地完成整个动画，应该说这个过程是漫长而艰苦的。不过万幸的是，今天我们使用的 SilverLight 技术可以代替完成这个工作。下面就开始介绍一下 SilverLight 中的动画技术。

在 SilverLight 中，动画对象是通过接收其各种属性的改变来实现的，如同我们刚刚介绍的动画原理，在 SilverLight 中，设定不同动画对象的属性状态，然后由计算机代替你完成拍照片、按暂停、调动作的循环操作，并体贴地把它们自动串接在一起，让用户最后看到的是一幅流畅的画面。

渐变动画

需要注意的一点是，在 SilverLight 中，对象接收作为动画效果修改的属性都必须使用 Double、Color、Point 作为值的属性，也就是说，其中的 Double 是使用各种数据，而 Color 则是用来显示颜色对象，Point 则多半用于位置信息。

下面我们来介绍一个 UI 元素如何实现动画，这个效果十分简单，就是实现一个矩形对象的淡入淡出效果。

要想实现矩形对象的淡入淡出效果，需要涉及 3 个对象，首先是矩形对象的属性，所有的 UI 元素都有一个 Opacity 属性，这个属性用于设定元素的透明度，取值是 0 到 1 之间的小数，其中 1 是完全不透明，0 是完全透明（关开属性更多的相关信息请参见图形绘制章节）。在这里还要提一下，如果要隐藏一个 UI 对象，最好的办法是设置 Visable 属性而不是把透明度设置为 0，因为后者的性能会低于前者。好了，这里要设置的属性就是 Opacity 属性，它是一个 Double 类型的属性，这些信息确定了之后，要创建第二个对象，即 DoubleAnimation，DoubleAnimation 指定一个 Double 类型的属性，使其在指定的时间内由起点值到达终点值，从而形成动画效果。应用它来实现动画效果，只要简单地指定几个参数值就可以了。这个对象是创建数据值类型渐变动画效果最重要的对象，在指定的 Duration 内使用线性内插对两个目标值之间的 Double 属性值进行动画处理。

动画在一段时间内更新属性值，动画的效果可能十分微小，比如将 Shape 左右移动几个像素；也可能十分显著，比如将对象放大到其原始大小的 200 倍，同时对其进行旋转并更改其颜色。若要在 Windows Presentation Foundation（WPF）中创建动画，请将动画与对象的属性值关联起来。

数据绑定控件

在 SilverLight 2.0 的诸多控件当中有一种控件非常常用，这就是数据绑定控件。

使用 DataGrid 可以方便灵活地显示数据，其单元格格式支持各种显示数据的方法，比如文本或者选择框，甚至还支持模板定义。DataGrid 控件支持一般的表格属性，比如支持单元格的背景色或者像一些表格的预置模板一样支持分行变色技术（这个技术被 ASP.NET 2.0 里的 GridView 发挥到了极致），此外还有隐藏表格、隐藏表线等多种表格常用技术。而且这个控件在设计的过程中希望完全支持用户的各种数据显示模式，所以预先内建了若干表格样式，用户可以方便地调用。下面就介绍如何在应用当中引入表格控件。

对于 DataGrid，最容易使用的方法就是通过指定它的 ItemsSource 属性来实现数据的显示。这个属性专门用于显示指定的数据，一旦指定了控件的 ItemsSource，控件就能自动生成各种表格数据用于显示。这里需要说明的一点是，一旦指定了 ItemsSource，数据的显示会自动把元数据中的数据类型转义成 String 的内容，而且这些文本在默认情况下是只读的。

如果希望使用自己定义的表格显示模式，则可以通过设置 AutoGenerateColumns 属性为 false 来限制自动生成表格内容，当然这也意味着你不得不通过使用 AutoGeneratingColumn 事件来自行处理所有的显示内容。如果希望对表格的显示内容进行排序，可以通过设置 DisplayIndex 属性来实现。

无论表格的内容是自动生成的还是编码指定的，它们都可以通过使用 DataGrid.Columns 对象来进行编程访问，可以调用它的 Add、Insert、Removeat 方法进行数据的添加、插入和删除。

刷子对象

在 SilverLight 中，色彩缤纷的颜色被封装成了一个对象，这个对象被用于绘制所有图形的边框、背景色以及其他填充效果。这个对象就是下面要介绍的刷子对象。

在前面的章节中我们已经见过不少刷子对象的例子，一般如同下面的样式：

```
<Ellipse Width="300" Height="100">
<Ellipse.Fill>
<SolidColorBrush Color="Red"/>
</Ellipse.Fill>
</Ellipse>
```

这种刷子对象定义了单一颜色的固体刷子，其实在 SilverLight 的刷子对象中，除了这个固体刷子之外，还有其他用于不同用途的刷子。在这一节里，我们会介绍 5 种常用的刷子对象，它们是 3 种基本图形填充刷子：

```
SolidColorBrush
LinearGradientBrush
RadialGradientBrush
```

以及 2 种多媒体刷子：

```
ImageBrush
VideoBrush
```

《软件开发这点事儿——软件开发工具手册》读者交流区

尊敬的读者：

感谢您选择我们出版的图书，您的支持与信任是我们持续上升的动力。为了使您能通过本书更透彻地了解相关领域，更深入的学习相关技术，我们将特别为您提供一系列后续的服务，包括：

1．提供本书的修订和升级内容、相关配套资料；

2．本书作者的见面会信息或网络视频的沟通活动；

3．相关领域的培训优惠等。

请您抽出宝贵的时间将您的个人信息和需求反馈给我们，以便我们及时与您取得联系。

您可以任意选择以下三种方式与我们联系，我们都将记录和保存您的信息，并给您提供不定期的信息反馈。

1．短信

您只需编写如下短信：B08072+您的需求+您的建议

发送到1066 6666 789（本服务免费，短信资费按照相应电信运营商正常标准收取，无其他信息收费）

为保证我们对您的服务质量，如果您在发送短信24小时后，尚未收到我们的回复信息，请直接拨打电话（010）88254369。

2．电子邮件

您可以发邮件至jsj@phei.com.cn或editor@broadview.com.cn。

3．信件

您可以写信至如下地址：北京万寿路173信箱博文视点，邮编：100036。

如果您选择第2种或第3种方式，您还可以告诉我们更多有关您个人的情况，及您对本书的意见、评论等，内容可以包括：

（1）您的姓名、职业、您关注的领域、您的电话、E-mail地址或通信地址；

（2）您了解新书信息的途径、影响您购买图书的因素；

（3）您对本书的意见、您读过的同领域的图书、您还希望增加的图书、您希望参加的培训等。

如果您在后期想退出读者俱乐部，停止接收后续资讯，只需发送“B08072+退订”至10666666789即可，或者编写邮件“B08072+退订+手机号码+需退订的邮箱地址”发送至邮箱：market@broadview.com.cn 亦可取消该项服务。

同时，我们非常欢迎您为本书撰写书评，将您的切身感受变成文字与广大书友共享。我们将挑选特别优秀的作品转载在我们的网站（www.broadview.com.cn）上，或推荐至CSDN.NET等专业网站上发表，被发表的书评的作者将获得价值50元的博文视点图书奖励。

我们期待您的消息！

博文视点愿与所有爱书的人一起，共同学习，共同进步！

通信地址：北京万寿路 173 信箱　博文视点（100036）　　电话：010-51260888

E-mail：jsj@phei.com.cn，editor@broadview.com.cn

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：（010）88254396；（010）88258888

传　　真：（010）88254397

E-mail：　dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

　　　　　电子工业出版社总编办公室

邮　　编：100036